Lappeenranta
University of Technology

Mohamad Ezral Baharudin

REAL-TIME SIMULATION OF MULTIBODY SYSTEMS WITH APPLICATIONS FOR WORKING MOBILE VEHICLES

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 2310 at Lappeenranta University of Technology, Lappeenranta, Finland on the 1st of April, 2016, at noon.

Acta Universitatis
Lappeenrantaensis 690

Supervisors Professor Aki Mikkola
LUT School of Energy Systems
Lappeenranta University of Technology
Finland

Associate Professor Marko Matikainen
LUT School of Energy Systems
Lappeenranta University of Technology
Finland

Reviewers Professor Zdravko Terze
Faculty of Mechanical Engineering and Naval Architecture
University of Zagreb
Croatia

Professor Taichi Shiiba
Department of Mechanical Engineering
Meiji University
Japan

Opponents Professor Zdravko Terze
Faculty of Mechanical Engineering and Naval Architecture
University of Zagreb
Croatia

Dr. Francisco Javier Gonzáles Varela
Mechanical Engineering Laboratory
University of La Coruña
Spain

ISBN 978-952-265-932-3
ISBN 978-952-265-933-0 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenrannan Teknillinen Yliopisto
Yliopistopaino 2016

Abstract

Mohamad Ezral Baharudin

Lappeenranta 2016
66 pages

Acta Universitatis Lappeenrantaensis 690
Diss. Lappeenranta University of Technology

ISBN 978-952-265-932-3
ISBN 978-952-265-933-0 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

This dissertation describes an approach for developing a real-time simulation for working mobile vehicles based on multibody modeling. The use of multibody modeling allows comprehensive description of the constrained motion of the mechanical systems involved and permits real-time solving of the equations of motion. By carefully selecting the multibody formulation method to be used, it is possible to increase the accuracy of the multibody model while at the same time solving equations of motion in real-time.

In this study, a multibody procedure based on semi-recursive and augmented Lagrangian methods for real-time dynamic simulation application is studied in detail. In the semi-recursive approach, a velocity transformation matrix is introduced to describe the dependent coordinates into relative (joint) coordinates, which reduces the size of the generalized coordinates. The augmented Lagrangian method is based on usage of global coordinates and, in that method, constraints are accounted using an iterative process.

A multibody system can be modelled as either rigid or flexible bodies. When using flexible bodies, the system can be described using a floating frame of reference formulation. In this method, the deformation mode needed can be obtained from the finite element model. As the finite element model typically involves large number of degrees of freedom, reduced number of deformation modes can be obtained by employing model order reduction method such as Guyan reduction, Craig-Bampton method and Krylov subspace as shown in this study

The constrained motion of the working mobile vehicles is actuated by the force from the hydraulic actuator. In this study, the hydraulic system is modeled using lumped fluid theory, in which the hydraulic circuit is divided into volumes. In this approach, the pressure wave propagation in the hoses and pipes is neglected. The contact modeling is divided into two stages: contact detection and contact response. Contact detection determines when and where the contact occurs, and contact response provides the force acting at the collision point. The friction between tire and ground is modelled using the LuGre friction model, which describes the frictional force between two surfaces.

Typically, the equations of motion are solved in the full matrices format, where the sparsity of the matrices is not considered. Increasing the number of bodies and constraint equations leads to the system matrices becoming large and sparse in structure. To increase the computational efficiency, a technique for solution of sparse matrices is proposed in this dissertation and its implementation demonstrated. To assess the computing efficiency, augmented Lagrangian and semi-recursive methods are implemented employing a sparse matrix technique. From the numerical example, the results show that the proposed approach is applicable and produced appropriate results within the real-time period.

Keywords: multibody system dynamics, real-time simulation, sparse matrix technique, semi-recursive method, augmented Lagrangian method

Acknowledgements

This work was carried out in the department of Mechanical Engineering at Lappeenranta University of Technology, Finland, between 2011 and 2016.

I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my supervisor, Professor Aki Mikkola. I have been amazingly lucky to have a supervisor who gave me freedom to explore on my own and guidance to recover when my steps faltered. His patience and support helped and inspired me to overcome many obstacles until I finished off this dissertation. I hope for one day that I will become as good as him in supervising my students as he has done to me. My thanks also to Associate Professor Marko Matikainen as a co-supervisor who always support and assist me along the way of my journey to complete this study.

Special thanks to my preliminary examiners Professor Zdravko Terze from University of Zagreb and Professor Taichi Shiiba from Meiji University for your valuable constructive comments on my dissertation. The opponents of the public examination, Professor Zdravko Terze from University of Zagreb and Dr. Francisco Javier Gonzáles Varela from University of La Coruña are also appreciated.

I am thankful to the current or former member of the Machine Design Laboratory whom I interacted with during my graduate studies and leisure time. This is the best team I ever worked with, in particular I would like to acknowledge Dr. Janne Heikkinen, Dr. Behnam Ghalamchi, Dr. Adam Klodowski, Dr. Scott Semken, Dr. Emil Kurvinen, John Bruzzo, Oskari Halminen, Jarkko Rahikainen, Vesa-Ville Hurskainen, Mojtaba Mobaraki, Dr. Tuomas Rantalainen, Dr. Antti Valkeapää, Dr. Oleq Dmitrochenko, and Elias Altariba for the many valuable discussions that helped me to understand my research area in a better way as well as about life. I will always remember every enjoyable moment we spent in our past secret coffee room. I am so grateful to the following staff at LUT, for their various forms of supports during my period of studies such as Jarkko Nokka, Sari Damsten, Eeva Häyrinen, Peter Jones, Anna-Kaisa Partanen, Johanna Jauhainen and Päivi Nuutinen. Thank you also to Dr. Asko Rouvinen and Dr. Pasi Korkealaakso from Mevea Ltd. for their assistance during the research cooperation was taking place.

Special thanks to my C programming’s guru, Azman Yusof and Associate Professor Abdul Rahman from Universiti Malaysia Perlis for your patience in providing assistance to me. Hopefully you will not feel annoying when answering all my outrageous questions.

When I came to Finland, I received overwhelming supports and assistance from Malaysian Students Association in Finland. My thanks to all members and the alumni especially to Dr. Azremi, Dr. Norsuria, Dr. Kamarizan & wife, Dr. Muzamir, Faizah, Dr.

Nurul Izza, Dr. Mohd Rafi, Normiza, Dr. Farizan, Zul Faizudin, Ummi Nor Nazahiah, Mohd. Azhan and Nurul Aida. My thanks also go to the current and former Malaysian embassy staff in Finland and their family, Mohd Zaky, Faizal, Maimunah Abdullah, Hussain Musa, Laimes Jali and Jerlene Wallenius for their outstanding supports.

Many friends have helped me stay sane through these tough years in Lappeenranta. Their supports and cares helped me a lot in overcoming obstacles and to stay in focused on my studies and family. I greatly value their friendship and I deeply appreciate their belief in me. Therefore, special thanks to Jussi Venalainen, Vina Ong, Jukka Heinonen and Roziyah Heinonen who helped me to adjust my life and family settling in this lovely country.

I also would like to say thank you to the people who always encouraged me to further my studies until to this PhD level. Firstly, deepest thanks to Datuk Prof. Emeritus Kamarudin Hussin and Datin Noridah Yangman for your trust and support when I started to develop my pathway in academic career. Special thank you also to Associate Professor Datuk Dr. Abu Bakar Mohd Diah and Datin Wan Faridah Wan Yaakob for their unwavering encouragement and motivation when I was at the bachelor degree.

Most importantly, none of this would have been possible without the love and patience of my family. My family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my lovely wife Norismiza Ismail for your patience and love through this journey. Becoming a wonderful mom to my two small kids, a hardworking master student, a lovely wife, a professional chef, best arguing partner and the best friend of me is surely not an easy job. For my two kids, Eryssa Nur Iman and Ezz Eilman, I hope one day you will understand how hard your dad had to work through to get to this level.

My sincere thanks to my parent and in law, Haji Baharudin Ahmad, Sofiah Che Embi, Haji Ismail Othman and Hajjah Umi Othman for your firm support and pray while we are thousand kilometres away from you. Also thanks to my sibling and in law, Ahmadi, Fitin Farhana, Mohd. Izwan, Noor Aniza, Mohd Hafizul, Nur A'alia, Mohd. Fadhli and Hasmiza for your support. My thanks also to all my relatives who always pray for my successfulness.

Finally, I would like to show my appreciation for the financial support to the Ministry of Higher Education of Malaysia and Universiti Malaysia Perlis who are funded my study here in Finland. My thanks also to Tampere Graduate School fund and SIMPRO research fund for partly supported my research.

Mohamad Ezral Baharudin
April 2016
Lappeenranta, Finland

Contents

Abstract

Acknowledgements

List of publications 9

Nomenclature 11

1 Introduction 15

1.1 Real-time simulation models 16
1.2 Real-time simulation environment 18
1.3 Outline of the dissertation 19
1.4 Scientific contribution 20

2 Real-time simulation methods for multibody system dynamics 23

2.1 Multibody system formulations 24
2.1.1 Kinematics 24
2.1.2 Equations of motion 27
2.1.3 Lagrange multiplier method 27
2.1.4 Penalty formulation 28
2.1.5 Augmented Lagrangian method 29
2.1.6 The semi-recursive method 30
2.1.7 The flexible body description 35
2.2 Hydraulic systems modeling 39
2.3 Collision and contact modeling 41
2.4 Friction modeling – The tire model 43

3 Sparsity in multibody methods 47

3.1 Sparsity in the equations of motion 47
3.2 Sparse solution methods 49

4 Programming libraries and numerical examples 51

4.1 Programming libraries 51
4.2 Numerical examples 53

5 Conclusions 59

5.1 Future work 60

References 61

Publications

List of publications

This thesis is based on the following papers. All papers are reprinted with permission of the publishers.

I. M. E. Baharudin, P. Korkealaakso, A. Rouvinen and A. Mikkola, "Crane operators training based on the real-time multibody simulation," in *Multibody Systems Dynamics, Robotics and Control*, Springer, pp. 213-229, 2013.

II. M. E. Baharudin, A. Rouvinen, P. Korkealaakso and A. Mikkola, "Tree harvester truck simulator based on real-time multibody simulation", in *ECCOMAS Thematic Conference on Multibody Dynamics 2013*, University of Zagreb, Croatia, pp. 359-374, 2013.

III. M. E. Baharudin, A. Rouvinen, P. Korkealaakso and A. Mikkola, "Real-time multibody application for tree harvester truck simulator," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics,* vol. 228, no. 2, pp. 182-198, 2014

IV. M. E. Baharudin, J. Nokka, J. H. Montonen, P. Immonen, A. Rouvinen, L. Laurila, T. Lindh, A. Mikkola, J. Sopanen, and J. Pyrhönen, "Simulation environment for real-time dynamic analysis of hybrid mobile machines", in *ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2015)*, Boston, Massachusetts, 2015.

V. M. E. Baharudin, A. Rouvinen, P. Korkealaakso, M. K. Matikainen, and A. Mikkola, "Real-time analysis of mobile machines using sparse matrix technique," in *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics,* SAGE. *Accepted on 3rd Feb. 2016.*

Author's contribution

The author is the corresponding author in all papers I – V. The articles were written under the supervision of Professor Aki Mikkola, who is a co-author in all the articles. As the author's supervisors, Professor Aki Mikkola and Associate Professor Marko Matikainen also supervised the writing of the author's dissertation.

In *Publication I*, the author was responsible for the compilation and structuring of the formulations used in the real-time simulator. Dr. Asko Rouvinen and Dr. Pasi Korkealaakso provided the numerical results from the simulator. The author wrote the manuscript, which was completed in co-operation with Professor Aki Mikkola.

In *Publication II & III*, the author wrote the formulations of the recursive method, actuator model and friction model. The numerical analysis was done by Dr. Asko Rouvinen and

Dr. Pasi Korkealaakso. The author wrote the manuscript, which was completed in co-operation with Professor Aki Mikkola.

In *Publication IV*, the author prepared the multibody formulation used in the simulator together with Professor Aki Mikkola. The author was also responsible for compiling all the formulations and the results from the other authors of the paper. The author was responsible for completing the final manuscript. This article is multidisciplinary and involves several different fields, namely, electrical drives technology, control engineering, machine dynamics and machine design.

In *Publication V (accepted)*, the author was responsible for preparing the multibody formulations together with Associate Professor Marko Matikainen. The author was also involved in the programming for the simulation. The numerical results were provided by Dr. Asko Rouvinen and Dr. Pasi Korkealaakso. The author wrote the manuscript, which was completed in co-operation with Associate Professor Marko Matikainen and Professor Aki Mikkola.

Nomenclature

Latin alphabet

$\mathbf{A}_i$	Rotation matrix at body i
$\mathbf{A}_{j\text{-}1,j}$	Relative rotation matrix between body j to body j-1
A	Surface area inside the hydraulic cylinder
A_b	Area behind cylinder piston
A_p	Area of cylinder piston
A_t	Cross section area of valve
$\mathbf{B}_c$	Constant force vector
B_{ei}	Effective bulk modulus at volume i
$\mathbf{b}$	General vectors at right hand side
$\mathbf{C}$	Cholesky factor
C	Damping coefficient
C_d	Flow coefficient
$\mathbf{D}_{j\text{-}1,j}$	Relative displacement vectors between point j to point j-1
$\mathbf{d}_p$	Distance between two points
d_p	Normal magnitude of contact
$\mathbf{F}_n$	Normal contact force
$\mathbf{F}_j$	External force on body j
F_A	Applied force to the tire hub
F_s	Force produced by hydraulic cylinder
F_μ	Friction force between piston and hydraulic cylinder wall
F_n	Normal force
$f(\dot{x})$	Velocity dependent coefficient of hydraulic cylinder
$\bar{\mathbf{G}}_i$	Local velocity transformation matrix between angular velocities and first derivative of Euler parameter of body i
$\dot{\bar{\mathbf{G}}}_i$	First time derivative of local velocity transformation matrix between angular velocities and first derivative of Euler parameter of body i
g_f	Static coefficient of friction
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Inertia tensor
$\mathbf{K}$	Stiffness matrix
$\mathbf{K}_{II}$	Stiffness matrix of internal degrees of freedom
$\mathbf{K}_{IB}$	Stiffness matrix of internal-boundary degrees of freedom
K	Stiffness coefficient
k	Iteration number
l	Maximum stroke of hydraulic cylinder
$\mathbf{M}$	Mass matrix
m_i	Mass of body i

m_o	Size of original dimension matrix in Krylov subspace
m_r	Size of reduction dimension matrix in Krylov subspace
$\mathbf{n}$	Normal vector of the contact
n_c	Number of constraint equations
n_d	Number of generalized coordinates
n_f	Number of degress of freedom
n_g	Number of generalized coordinates
n_h	Number of pipes and hoses
n_m	Number of modes
n_p	Number of points used to describe a flexible body
P	Point on the rigid body
$\mathbf{p}$	Vectors of modal coordinates
p_1	Pressure behind piston side in the hydraulic cylinder chamber
p_2	Pressure on piston rod side in the hydraulic cylinder chamber
p_{in}	Hydraulic input pressure
p_{out}	Hydraulic output pressure
p_T	Pressure at tank
p_P	Pressure at pump
$\dot{p}_i$	Differential pressure at volume *i*
$\mathbf{Q}_c$	Vector of velocity dependent terms due to differentiation of constraint equations
$\mathbf{Q}_e$	Vector of generalized forces
$\mathbf{Q}_v$	Vector of quadratic velocity in inertia terms
$Q_{in,i}$	Incoming flowrate to volume *i*
$Q_{out,i}$	Outgoing flowrate from volume *i*
Q_t	Flowrate in the restrictor valve
$\mathbf{q}$	Vectors of generalized coordinates
$\mathbf{q}_i$	Vectors of generalized coordinates of body *i*
$\dot{\mathbf{q}}$	Vectors of generalized velocities
$\ddot{\mathbf{q}}$	Vectors of generalized accelerations
$\mathbf{R}_i$	Position of body reference coordinate system
$\dot{\mathbf{R}}_i$	Velocity of body reference coordinate system
$\ddot{\mathbf{R}}_i$	Acceleration of body reference coordinate system
$\mathbf{R}^{cm}$	Position of body reference coordinate system at the center of mass
$R_{i\mathrm{X}}$	Vector of component-X defining the body reference coordinate of body *i*
$R_{i\mathrm{Y}}$	Vector of component-Y defining the body reference coordinate of body *i*
$R_{i\mathrm{Z}}$	Vector of component-Z defining the body reference coordinate of body *i*
$\mathbf{r}_{iP}$	Vectors of position of point *P* at body *i* in global system
$\dot{\mathbf{r}}_{iP}$	Vectors of velocity of point *P* at body *i* in global system
$\ddot{\mathbf{r}}_{iP}$	Vectors of acceleration of point *P* at body *i* in global system
r_W	Radius of tire

$\mathbf{T}$	Vector of torsional forces
t	Time
$\mathbf{u}_{\mathrm{d}}$	Direction vector pointing to the direction of joint
$\bar{\mathbf{u}}_{iP}$	Vectors of position of point P with at body i respect to body reference coordinate
$\tilde{\bar{\mathbf{u}}}_{iP}$	Skew symmetric matrix of position vector of point P
$\bar{\mathbf{u}}_{j-1}$	Vector of position within body reference coordinate system of body j-1
$\bar{\mathbf{u}}_{f}$	Displacement vectors of deformation
$\bar{\mathbf{u}}_{o}$	Displacement vectors of undeformed position
$\mathbf{V}$	Velocity transformation matrix
$\dot{\mathbf{V}}$	First time derivative of velocity transformation matrix
V_i	Total volume i
$\dot{V}_i$	Change of volume i with respect to time
v_n	Relative normal velocity of contact
v_r	Relative velocity between two sliding surfaces
v_s	Stribeck relative velocity
v_v	Vehicle velocity
W_{iner}	Work of inertia forces
W_{ext}	Work of external forces
$\mathbf{W}$	Square matrix
$\mathbf{x}$	General vectors at left hand side
x	Displacement of hydraulic piston
$\dot{x}$	Velocity of hydraulic piston
$\mathbf{y}$	General vectors
$\mathbf{z}$	Vectors of relative position coordinates of joint
$\dot{\mathbf{z}}$	Vectors of relative velocities of joint
$\ddot{\mathbf{z}}$	Vectors of relative accelerations of joint
z	Bristle displacement in Lugre friction model

Greek alphabet

$\boldsymbol{\theta}_{iE}$	Rotational Euler parameters
$\dot{\boldsymbol{\theta}}_{iE}$	First time derivative of rotational Euler parameters
$\theta_0, \theta_1, \theta_2, \theta_3$	Euler parameters
$\bar{\boldsymbol{\omega}}_i$	local angular velocity of body i
$\tilde{\bar{\boldsymbol{\omega}}}_i$	Skew-symmetric matrix of local angular velocity of body i
$\dot{\boldsymbol{\omega}}_i$	Local angular acceleration of body i
$\boldsymbol{\omega}_n$	Natural frequencies
ω_w	Tire angular velocity
$\boldsymbol{\Phi}$	Kinematic constraint equations
$\dot{\boldsymbol{\Phi}}$	First time derivative of kinematic constraint equations

$\ddot{\mathbf{\Phi}}$	Second time derivative of kinematic constraint equations
$\mathbf{\Phi}_{cb}$	Craig-Bampton transformation
$\mathbf{\Phi}_{B}$	Static modes
$\mathbf{\Phi}_{d}$	Deformation modes
$\mathbf{\Phi}_{g}$	Reduction transformation matrix of Guyan reduction method
$\mathbf{\Phi}_{I}$	Fixed boundary mode
$\mathbf{\Phi}_{k}$	Reduction transformation matrix of Krylov subspace method
$\mathbf{\Phi}_{\mathbf{q}}$	Jacobian of kinematic constraint equations
$\dot{\mathbf{\Phi}}_{\mathbf{q}}$	First time derivative of Jacobian of kinematic constraint equations
$\mathbf{\Phi}_{\mathbf{z}}$	Jacobian of constraint with respect to variable $\mathbf{z}$
$\boldsymbol{\lambda}$	Lagrange multiplier
$\boldsymbol{\alpha}$	Penalty term
ξ	Damping ratio coefficient
ε	Tolerance
$\mathbf{\Omega}$	Eigenvectors
ρ	Density of fluids
η	Hydraulic cylinder efficiency
μ_c	Normalize Coulomb friction
μ_s	Normalize static fricition
σ_0	Rubber longitudinal lumped stiffness
σ_1	Longitudinal lumped damping coeffcient
σ_2	Viscous relative damping

Abbreviations

AABB	Axis-Aligned Bounding Box
ARPACK	Arnoldi Package
BLAS	Basic Linear Algebra Subprograms
CG	Conjugate Gradient
CSC	Compressed Sparse Column format
DAE	Differential Algebraic Equation
GPL	General Public Licenses
HB	Harwell Boeing sparse format
LAPACK	Linear Algebra Package
NCSS	Normet Concrete Spray System
OBB	Oriented Bounding Box
ODE	Ordinary Differential Equation
SuperLU	Sparse Linear Equation Solver

1 Introduction

Computer simulation has proved to be a very effective tool when designing machines. Compared to previous methods, which were often based on analytic solutions or empirical testing, computer simulation offers remarkable improvements in prediction of machine performance. A key advantage of computer simulation of machines is that it enables fast and effective study of the effects of design variables on the dynamic behavior. As a consequence, computer simulation is currently implemented as an integral part of many industrial design processes. The use of computer simulation reduces the need to build physical prototypes, thus accelerating the product development cycle. When considering the dynamic performance of machines, it is important to note that the experience level of operators plays a critical role. However, most of the simulation research has been focused on the development of modeling methods of machine components while studies that account the influence of the operator have been with little attention. Generally, this is due to the fact that there is no mathematical expression that describes the behavior of operators. Operators can be taken into account by employing real-time simulation as depicted Figure 1.1.

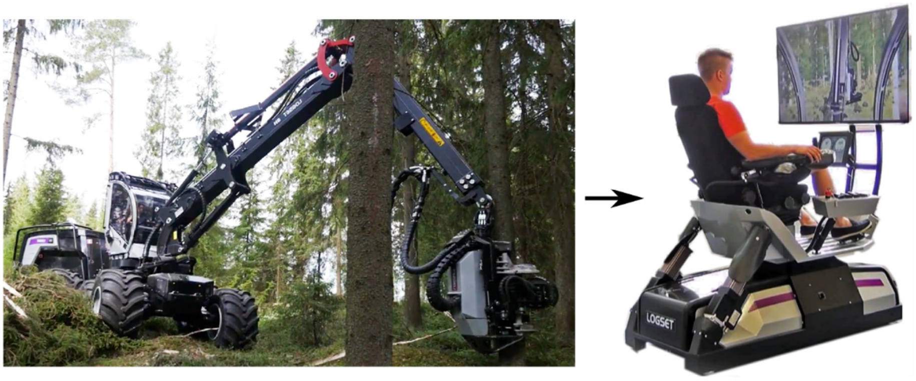


Figure 1.1: Transformation from on-site training to real-time simulator (adapted with permission from Mevea Ltd.)

In real-time simulations, the operator is actively engaged in the dynamic performance of the machine under simulation. Using real-time simulation, operators can take an active part in early phases of the machine design process, and operators can experience the performance of the machine long before the first prototype is built. This is an important issue since the dynamic performance of a machine is often a matter of feeling rather than explicit mathematical expression. Interaction between the operator and simulation model also provides an opportunity to utilize simulations in user training.

Simulation based user training provides a number of advantages compared to traditional training, for example, the possibility of giving training and knowledge of the machine to be operated and the operation environment without the requirement for a real machine. When using a real-time simulator, existing machine capacity is not tied to training activities and can be used in productive work. Furthermore, using a simulator means that accidents resulting from inexperience, which might cause damage to operators, the machine or the environment, can be avoided. Additionally, simulators enable operators to be trained for accident scenarios. A further advantage is that all operators can receive training in a variety of different environmental conditions, such as different lighting conditions, fog and wind.

In order to assess the performance of a machine using computational methods, the system dynamics must be solved. A number of studies on how to solve the system dynamics of machines can be found in the literature [1] [2] [3]. In this study, a multibody system dynamics is used to model dynamic systems.

1.1 Real-time simulation models

In this study, the complete real-time simulation model are considered as interaction of several subsystems as depicted in Figure 1.2.

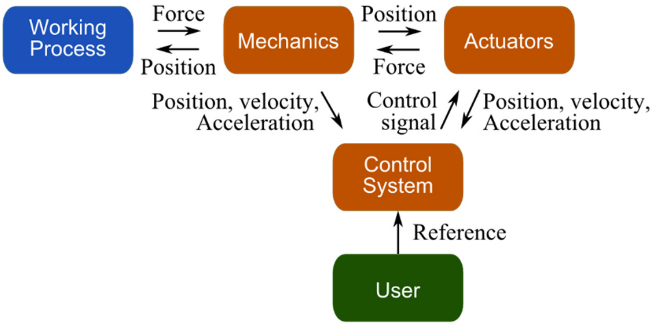


Figure 1.2: Subsystem and communication in real-time simulation model

In Figure 1.2, each subsystem interact with other subsystem in order to provide complete simulation process. The main role of the user is to provide reference signal to the control system via a control console (pedal, joystick etc.), which can be translated as input signals for the control system. Control system is where most of the input/output data is processed and synchronized with the other subsystems. The actuators subsystem is functional to compute the force need to be provided to the mechanic subsystem after receiving data signal from the control system. For example, the actuators, which are assumed to be hydraulic actuators in this work, produce forces on the mechanical system, which are transformed into the constrained motion. The constrained motion is computed in the mechanics subsystem where the selected multibody formulations are used.

In this study, description of mechanical subsystem is based on the use of multibody system dynamics and it includes modeling of mechanical bodies, contacts, and tires. In multibody approach, the set of position coordinates can be defined using generalized global coordinates or relative coordinates [4] [5]. The set of coordinates is also used to define the velocity and acceleration of bodies in the system. In the global formulation, the generalized global coordinates represent the position and orientation of all bodies in the system. The kinematics of the joint is derived based on the constraint equations [6]. The relative coordinates are used in the topology method, where the kinematics of the body is obtained recursively through the open loop system. This approach often leads to a small number of generalized coordinates. However, a closed loop system must be opened and the kinematic constraints at the removed joint must be accounted for in the dynamic solution [7]. To express the equations of motion, the dynamic equilibrium of the system must be defined. This equilibrium can be obtained by employing an approach such as the principle of Virtual Work [8]. A multibody system is a constraint system and thus the constraints need to be considered when defining the equations of motion. Selection of the formulation that accounts the constraints is crucial. Several formulations are available such as velocity transformation, penalty method, and augmented Lagrangian method as will be explained in this study [9].

In the case of flexible bodies, a floating frame of reference formulation can be used [10]. In this method, the configuration of the flexible bodies is defined using transformation and rotation of the reference coordinate system as well as body deformation relative to the reference coordinate system [5] [11]. The body deformation can be described using a deformation modes, which can be obtained using finite element methods. Finite element methods will normally offer a large number of nodal coordinates, which can lead to high number of deformation modes and, consequently, high computational costs if used in multibody applications. This drawback can be addressed by the use of model order reduction methods [12] [13]. Several methods can be implemented, such as Guyan reduction, the Craig-Bampton method and the Krylov subspace as will be explained in this work.

The equations of motion can be integrated using the explicit Runge-Kutta method with a fixed time step. The most widely used in this method is the fourth order (RK4) method and in general, it is used to solve a first-order system of ordinary differential equation. This method require four functions evaluations per time step which make it easy to be implemented [14]. This method is claimed so accurate and favourite solution for differential equation [15].

Contact modeling can be divided into two stages: collision detection and collision response modeling [16]. The collision detection model determines when and where the collision will occur. The collision response model prevents penetration between two bodies and provides the reacting contact force at the contact point.

In this work, dynamic tire friction is modelled using a lumped LuGre friction model, proposed by Canudas de Wit, to model the friction between the two surfaces involved (the ground and tires). The contact point between two surfaces is represents as bristle at the microscopic level. The friction model is based on elastic deformation of the bristle, which is called the Stribeck effect [17]. When a tangential force is applied to the bristle, it deflects like a spring and the amount of deflection displacement can be described in terms of the friction force. Using this friction model, many important aspects of friction can be accounted for, including stiction, the Stribeck effect, stick slip, zero slip displacement and hysteresis.

The hydraulic system model describes the actuators subsystem. It can be modelled using lumped fluid theory, where the hydraulic circuit is divided into discrete volumes with an assumption that the pressure is distributed equally [18]. The hydraulic system normally has a high nominal frequency, so the time step must be smaller to obtain a reliable result and hydraulic stability.

1.2 Real-time simulation environment

Real-time simulation environment is an application that can be seen as a combination of the virtual and the real world as an example, the training simulator. Real-time simulation requires knowledge of computational dynamics to transform the physical machine into a computer model that can imitate actual behavior of a dynamical system [19]. Real-time simulation environments often consist of a number of distinct features such as the visualization system, monitoring system, motion system and cockpit, as depicted in Figure 1.3.

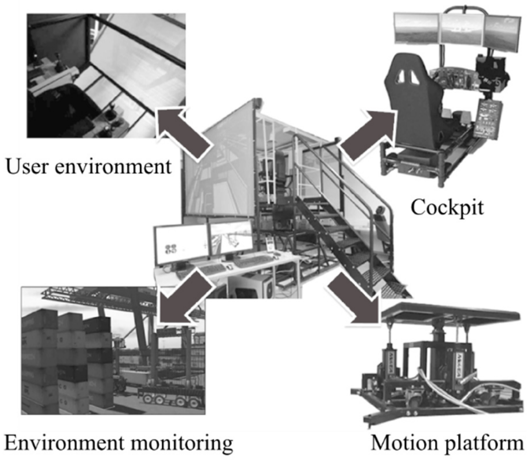


Figure 1.3: Real-time simulation environment

A realistic real-time simulator must feel like usage of a real machine, which is only achievable if the real-time simulation model is physics based and accurate, and if it combines realistically the different engineering areas involved. When using a real-time simulator the operator sees a virtual world moving on the visualization system. The surrounding sounds provide the aural environment, giving the simulator further realism.

In addition to the clearly essential visual and audio feedback, feedback from the motion platform is important aspect of operator training. If the operator does not feel the accelerations of the machine system caused by control maneuvers, it is possible that the training will be suboptimal and result in too fast and harsh operating behavior, since such maneuvers seem not cause any physical feedback. If, however, motion platform feedback is added, the operating behavior becomes smoother because hasty maneuvers cause unpleasant accelerations and operators thus correct undesirable behavior. Real-time simulation environments couple sophisticated mathematical models, visualization, sound and a motion platform to one complex system.

1.3 Outline of the dissertation

The objective of this dissertation is to study available multibody methods in a real-time simulation application. The main goal of this thesis is to show how the multibody model interacts with the selected method in different applications in terms of performance and efficiency. Comparison between the multibody formulations used in the simulation process enables identification of the suitability of the multibody model with the selected method.

This thesis is divided into five chapters. Following the introduction to the work in chapter one, chapter two discusses the real-time multibody methods used in the simulation process. These methods include the Lagrange multiplier method, penalty formulation method, augmented Lagrangian method, recursive method and description of flexible bodies. The description of the flexible bodies forms the basis for the kinematics analysis and the deformation mode to describe the translation of deformation coordinates is discussed with the model order reduction methods possibilities. Chapter three discusses sparsity in the system matrices in the multibody system model and explores possibilities to increase the efficiency of the numerical algorithm when a sparse technique is implemented. Chapter four describes the programming approach and presents some numerical examples to identify the performance and efficiency of the model. The conclusions of the study are summarized in Chapter five, where suggestions for future work are also given.

1.4 Scientific contribution

This dissertation discusses multibody, hydraulic, contact, and tire modeling methods that can be used in real-time simulation. The approaches introduced are applied for several mobile vehicles such as a gantry crane, tree harvester tractor, mining loader, and mining concrete spray vehicle. The results of these studies can be extended to other working mobile vehicles for product development or training purposes. The scientific contributions can be summarized in three categories as follows.

First, this study applies two multibody formulations to simulate the dynamic behaviors of a working mobile vehicle. The first is based on an augmented Lagrangian method, and the second is a semi-recursive formulation. Both are applied separately to solve the system as a dynamic system of rigid bodies in a real-time simulation. The computational efficiencies of both methods are compared. In certain cases, the semi-recursive method provides a significant computational efficiency advantage over the Lagrangian method. Comparing these two simulation approaches to solve this type of problem has not been previously described, and the results presented here represent new scientific information.

Second, a flexible body description is formulated based on a floating frame of reference formulation. In this formulation, the deformation modes, which can be obtained from the finite element method, are calculated by employing model order reduction methods. Three methods are used: the Guyan reduction method, the Craig-Bampton method, and Krylov subspace method. The reduction of the deformation modes helped to increase computational efficiency while producing acceptable real-time simulation results. To author knowledge, this approach to the selection of reduction methods specifically for real-time application has not been described previously in the literature. Therefore, this represents new scientific information for the real-time simulation community.

Third, increasing the number of bodies in a multibody system expands the dimensions of the matrices in the equations of motion. However, each matrix contains fewer non-zero elements, and the matrix becomes sparse. To address this sparsity, a sparse matrix approach is introduced as an option to solve the equations of motion used in real-time simulation applications. This approach is taken to better comprehend the effect of computing efficiency in the real-time simulation. The sparse technique is implemented with the augmented Lagrangian and semi-recursive methods to identify possible computational efficiency improvement. The results obtained from the simulation for both methods have not been described previously, so the results represent new scientific information.

Fourth, a multibody model is coupled with sub-models, e.g., a hydraulic model, a contact model, and a tire friction model to produce a realistic working mobile vehicle simulation. The hydraulic model is developed based on lumped-fluid theory. Contacts are modeled using a penalty method where spring-damper elements are added at the contact point to describe contact force. A lumped LuGre friction model is used to determine the friction force between the tire and the ground. Even though some of the sub-models are modeled using simple physics relations, the results are acceptable and realistic and determined within the real-time period. To author knowledge, combining these sub-models for a real-time simulation application has not been described in the literature, and this represents new scientific information.

2 Real-time simulation methods for multibody system dynamics

Multibody system dynamics offers a straightforward computer-based approach to generating and solving equations of motion for complex mechanical systems. It can be traced back to 1960’s when the computer simulation of satellites was employed. Multibody system can be seen as a system of bodies, which are kinematically constrained with a different types of joints [5]. The kinematics and dynamics of a multibody system can be solved using a number of approaches. All the approaches have their own advantages and drawbacks, depending on their suitability to the type of model to be implemented. In real-time simulation, the selected method must produce accurate results robustly and efficiently. Therefore, selection of the method to be used to solve the kinematics and dynamics is crucial to ensure the results can be produced within the real time period.

As a multibody system is a constrained system, the equations of motion are discussed based on several methods, namely, the Lagrange multiplier method, penalty method, augmented Lagrangian method and semi-recursive method. The flexible bodies can be described using a floating frame of reference formulation with model order reduction methods, which are also discussed in this section.

In a mobile working vehicle modeling, besides of multibody system dynamics, also involves description of hydraulic, contact and tire subsystems, as shown in Figure 2.1.

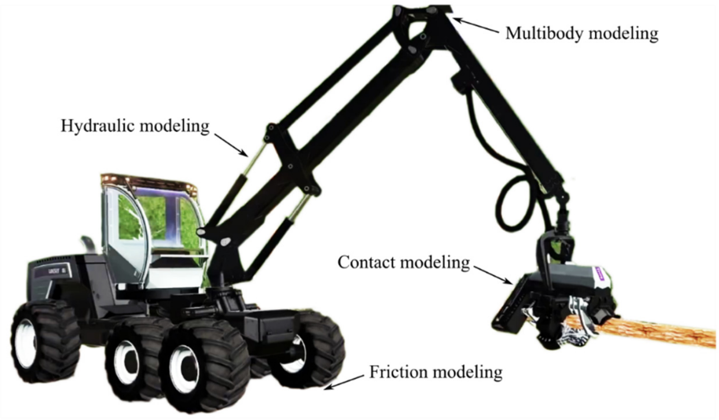


Figure 2.1: Modeling tasks involved in the real-time simulation (adapted with permission from Mevea Ltd.)

The hydraulic system provides actuating forces to the multibody system. The contact model, in turn, is needed to characterize the collision between two bodies, such as between the gripper and log, or tire and ground. In the case of a working mobile vehicle, the friction between the tire and ground also need to be modeled in order to describe friction between tire and terrain.

2.1 Multibody system formulations

The equations of motion can be solved using numerical time integration methods once the differential equations have been developed. However, before the equations of motion can be developed, the kinematics of the multibody system needs to be determined. The dynamic equilibrium of the multibody system can be determined using the concept of virtual work by including constraint equations.

In this study, the kinematics system is described using global formulation and a topological method. In the global formulation, the coordinates of the bodies are described with respect to the global frame of reference. In the topological method, in turn, the position of the body is based on the preceding body reference frame, as will be explained in the description of the semi-recursive method.

2.1.1 Kinematics

In the global formulation, the global position $\mathbf{r}_{iP}$ of point *P* on a rigid body *i*, as shown in Figure 2.2, can be described using a body reference coordinate system as follows:

$$\mathbf{r}_{iP} = \mathbf{R}_i + \mathbf{A}_i \bar{\mathbf{u}}_{iP} \tag{2.1}$$

where the vector $\mathbf{R}_i$ defines the position of the body reference coordinate system with respect to the global coordinate system XYZ; matrix $\mathbf{A}_i$ is the rotation matrix defining the orientation of the body reference coordinate system with respect to the global coordinate system; and vector $\bar{\mathbf{u}}_{iP}$ defines the position of point *P* with respect to the body reference coordinate system xyz.

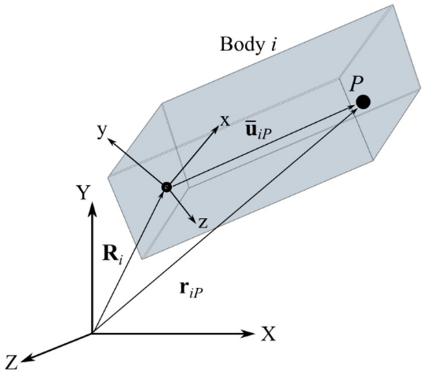


Figure 2.2: Description of position in the body reference coordinate system xyz and global coordinate system XYZ

In the three dimensional systems, the origin of the body reference coordinate can be defined by $\mathbf{R}_i = \begin{bmatrix} R_{iX} & R_{iY} & R_{iZ} \end{bmatrix}^{\mathrm{T}}$ and its orientation by a vector of rotational coordinates $\boldsymbol{\theta}_{iE}$. Therefore, the vector of the generalized coordinates $\mathbf{q}_i$ of a rigid body i can be written as

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{R}_i^{\mathrm{T}} & \boldsymbol{\theta}_{iE}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{2.2}$$

The rotational coordinates $\boldsymbol{\theta}_{iE}$ can be described using Euler parameters as:

$$\boldsymbol{\theta}_{iE} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^{\mathrm{T}} \tag{2.3}$$

where θ_0, θ_1, θ_2, and θ_3 are called Euler parameters as described in [5]. Using Euler parameters, the rotation matrix $\mathbf{A}_i$ can be expressed as:

$$\mathbf{A}_i = \begin{bmatrix} 1-2\theta_2^2-2\theta_3^2 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 1-2\theta_1^2-2\theta_3^2 & 2(\theta_2\theta_3+\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 1-2\theta_1^2-2\theta_2^2 \end{bmatrix}_i \tag{2.4}$$

When applying Euler parameters, the following mathematical constraint must be fulfilled:

$$\boldsymbol{\theta}_{iE}^{T}\boldsymbol{\theta}_{iE} - 1 = 0 \tag{2.5}$$

The velocity of point P on a rigid body i, $\dot{\mathbf{r}}_{iP}$ can be obtained by differentiating equation (2.1) with respect to time as

$$\begin{aligned}\dot{\mathbf{r}}_{iP} &= \dot{\mathbf{R}}_i + \dot{\mathbf{A}}_i \bar{\mathbf{u}}_{iP} \\ &= \dot{\mathbf{R}}_i - \mathbf{A}_i \tilde{\bar{\mathbf{u}}}_{iP} \bar{\boldsymbol{\omega}}_i\end{aligned} \tag{2.6}$$

where $\dot{\mathbf{R}}_i$ is the first time derivative of position vector of the origin of the body reference coordinate, $\dot{\mathbf{A}}_i$ is the first time derivative of its rotation matrix, $\tilde{\bar{\mathbf{u}}}_{iP}$ is the skew-symmetric matrix of vector $\bar{\mathbf{u}}_{iP}$ and $\bar{\boldsymbol{\omega}}_i$ is the vector of local angular velocities. The angular velocity vector $\bar{\boldsymbol{\omega}}_i$ can be obtained using the time derivative of the Euler parameters as:

$$\bar{\boldsymbol{\omega}}_i = \bar{\mathbf{G}}_i \dot{\boldsymbol{\theta}}_{iE} \tag{2.7}$$

where $\dot{\boldsymbol{\theta}}_{iE}$ is the first time derivative of vector of Euler parameters and $\bar{\mathbf{G}}_i$ is the local transformation matrix dealing with local and global component of body *i* which can be expressed in Euler parameters as:

$$\bar{\mathbf{G}}_i = \begin{bmatrix} -\theta_1 & \theta_0 & \theta_3 & -\theta_2 \\ -\theta_2 & -\theta_3 & \theta_0 & \theta_1 \\ -\theta_3 & \theta_2 & -\theta_1 & \theta_0 \end{bmatrix}_i \tag{2.8}$$

Therefore, the velocity vector $\dot{\mathbf{r}}_{iP}$ can be written in terms of generalized velocities $\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{\mathbf{R}}_i & \dot{\boldsymbol{\theta}}_{iE} \end{bmatrix}^{\mathrm{T}}$ as:

$$\dot{\mathbf{r}}_{iP} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}_i \tilde{\bar{\mathbf{u}}}_{iP} \bar{\mathbf{G}}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}}_i \\ \dot{\boldsymbol{\theta}}_{iE} \end{bmatrix} \tag{2.9}$$

where $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. The acceleration vector $\ddot{\mathbf{r}}_{iP}$ can be obtained from the derivative of equation (2.9) with respect to time as:

$$\ddot{\mathbf{r}}_{iP} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}_i \tilde{\bar{\mathbf{u}}}_{iP} \bar{\mathbf{G}}_i \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}}_i \\ \ddot{\boldsymbol{\theta}}_{iE} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{A}_i \tilde{\bar{\boldsymbol{\omega}}}_i \tilde{\bar{\mathbf{u}}}_{iP} \bar{\mathbf{G}}_i - \mathbf{A}_i \tilde{\bar{\mathbf{u}}}_{iP} \dot{\bar{\mathbf{G}}}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}}_i \\ \dot{\boldsymbol{\theta}}_{iE} \end{bmatrix} \tag{2.10}$$

where generalized accelerations $\ddot{\mathbf{q}}_i = \begin{bmatrix} \ddot{\mathbf{R}}_i & \ddot{\boldsymbol{\theta}}_{iE} \end{bmatrix}$, $\ddot{\mathbf{R}}_i$ is the second time derivative of of position vector of the origin of the body reference coordinate, $\ddot{\boldsymbol{\theta}}_{iE}$ is the second time derivative of vector of Euler parameters, $\tilde{\bar{\boldsymbol{\omega}}}_i$ is the skew-symmetric matrix of angular velocity and $\dot{\bar{\mathbf{G}}}_i$ is the first time derivative of the local transformation matrix.

2.1.2 Equations of motion

Equations of motion can be developed using the principle of least action and the concept of virtual work. The dynamic equilibrium for an unconstrained system can be obtained by equalizing the virtual work done by external forces with the virtual work done by inertial forces as:

$$\delta W_{\text{iner}} = \delta W_{\text{ext}} \tag{2.11}$$

where δW_{iner} is the virtual work of the inertia forces and δW_{ext} is the virtual work of the externally applied forces. The virtual work of the inertia and externally applied forces can be, respectively, written as [4],

$$\delta W_{\text{iner}} = \delta \mathbf{q} \cdot (\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_v) \tag{2.12}$$

$$\delta W_{\text{ext}} = \delta \mathbf{q} \cdot \mathbf{Q}_e \tag{2.13}$$

where $\mathbf{M}$ is the mass matrix, $\ddot{\mathbf{q}}$ is the generalized accelerations, $\mathbf{Q}_v$ is the vector of quadratic velocity vector and $\mathbf{Q}_e$ is the vector of generalized forces of the multibody system. Using equations (2.12) and (2.13), the equation can be written as

$$\delta \mathbf{q} \cdot (\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_v - \mathbf{Q}_e) = \mathbf{0} \tag{2.14}$$

However, in a multibody system the term $(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_v - \mathbf{Q}_e)$ cannot be set as equal to zero because the constraints of system are not taken into consideration. Therefore, the dynamic equilibrium in multibody applications can be obtained by introducing constraints into the equations of motion, as will be discussed in the next section.

2.1.3 Lagrange multiplier method

In multibody system, the constraint equations need to be coupled into the equations of motion. The constraints can be represented using a vector of constraints $\mathbf{\Phi}$. The vector of constraints consists of the number of constraint equations, n_c which should be less or equal to the number of generalized coordinates, n_g. To satisfy the constraint equations related to the generalized coordinates, the following equation must be fulfilled:

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \tag{2.15}$$

where t is time. The idea of the Lagrange multiplier method is to introduce constraints into the equations of motion by means of a set of multipliers, $\boldsymbol{\lambda}$. The vector of the Lagrange multipliers is related to the reaction forces due to the constraints. The vector of

Lagrange multipliers can be selected such that the following equation can be set as equal to zero as follows:

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_e - \mathbf{Q}_v + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{2.16}$$

where $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraints. By taking into account equation (2.15) and (2.16), the equations of motion can be written as

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_e - \mathbf{Q}_v + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \\ \mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \end{cases} \tag{2.17}$$

Equation (2.17) is a differential algebraic equation (DAE), which can be solved by converting it to an ordinary differential equation (ODE). This can be carried out by differentiating the vector of constraints (2.15) twice with respect to time as follows:

$$\mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt} \tag{2.18}$$

where $-(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}$ can be represents as $\mathbf{Q}_c$, $\mathbf{\Phi}_{\mathbf{q}t}$ is the first time derivative of Jacobian matrix of constraints, and $\mathbf{\Phi}_{tt}$ second time derivative of the vector of constraints. In the case of the scleronomic system, the time dependent constraints can be eliminated from the equation (2.18). Considering equations (2.17) and (2.18), the equations of motion can be formed into a matrix format as:

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_e + \mathbf{Q}_v \\ \mathbf{Q}_c \end{bmatrix} \tag{2.19}$$

In this equation, the acceleration vectors $\ddot{\mathbf{q}}$ and Lagrange multipliers $\boldsymbol{\lambda}$ are unknowns. It should be noted that equation (2.19) only represents the constraints in terms of acceleration. When differentiating the constraint equations twice with respect to time, the constant components in the constraint equations disappear. This value must be identified in order to keep the constraint equations fulfilled at any given time according to equation (2.15). This drawback can be alleviated by introducing a constraint stabilization method such as the Baumgarte stabilization method [20], a penalty formulation or an augmented Lagrangian method among others. In following sections, the penalty formulation and augmented Lagrangian method will be described.

2.1.4 Penalty formulation

The penalty method introduces penalty terms into the equations of motion, as suggested in [21]

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_e + \mathbf{Q}_v - \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{2.20}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\omega}_n$ and $\boldsymbol{\xi}$ are diagonal matrices contain the values of penalty factor, natural frequencies and damping ratios, respectively, at each constraint condition. When penalty value $\boldsymbol{\alpha}$ is set to large, it will ensure the constraints within tight tolerances but might lead to the numerical problems. The value selection for $\boldsymbol{\xi}$ and $\boldsymbol{\omega}_n$ can be varied as no general procedures are available to determine their values. Therefore, this approach may produce inconsequential results in certain cases [22].

2.1.5 Augmented Lagrangian method

The augmented Lagrangian method was developed to overcome the drawbacks of the penalty method and the Lagrange multiplier method. As mentioned earlier, large penalty values will converge the constraints within a tight tolerance while it may lead to numerical illness and round-off error [4]. In the augmented Lagrangian method, an iterative procedure is introduced to account the constraints in the equations of motion.

In this method, equation (2.20) can be augmented by adding the Lagrange multipliers. The equations of motion can then be written as

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q}_e + \mathbf{Q}_v - \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{2.21}$$

where $\boldsymbol{\lambda}$ is the Lagrange multipliers governing the new stable equation. As the Lagrange multipliers are steady enough to impose the constraints in the equations, the numerical value of the penalty coefficient does not need to be large [22]. In this method, Lagrange multipliers are not assumed as unknowns but computed through an iterative process as:

$$\begin{gathered}\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \boldsymbol{\alpha}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi})_{k+1} \\ k = 0,1,2,\ldots\end{gathered} \tag{2.22}$$

where subscript k is iteration number, and initial $\boldsymbol{\lambda}_0 = 0$. As the iterative solution solves the Lagrange multipliers, the augmented Lagrangian formulation leads to a system of ordinary differential equations without additional unknowns. The system of equations of motion, including the iterative scheme, can be written in the form of index-1 as follows:

$$(\mathbf{M} + \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}})\ddot{\mathbf{q}}_{k+1} = \mathbf{M}\ddot{\mathbf{q}}_k + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}(\boldsymbol{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{tt} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{2.23}$$

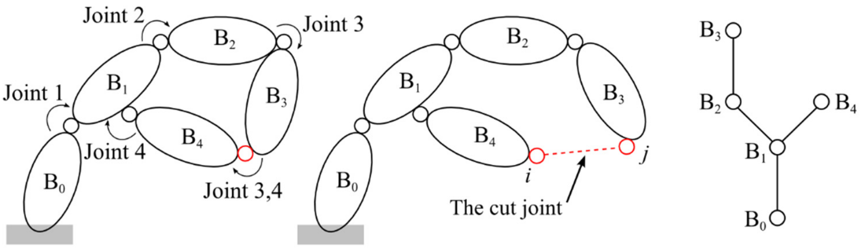
where for initial iteration, $\mathbf{M}\ddot{\mathbf{q}}_{k=0} = \mathbf{Q}_e + \mathbf{Q}_v$. The iterative process is repeated until the setting tolerance ε meets the form:

$$\left\| \ddot{\mathbf{q}}_{i+1} - \ddot{\mathbf{q}}_i \right\| = \varepsilon \tag{2.24}$$

As this procedure involves an iteration loop, it may lead to extra computing time. The procedure is normally implemented by employing Newton-Raphson iteration method [23]. A major advantage of using this approach is that the penalty factor values are no longer critical because they are treated by the Lagrange multipliers.

2.1.6 The semi-recursive method

In this method, the kinematic properties such as position, velocity and acceleration are derived based on the relative coordinates between contiguous bodies connected by a joint. The dynamics equation in this method is written in terms of the system's degrees of freedom and, typically, is written with a lower dimensionality than those in an augmented formulation. In this method, any closed loop systems will be treated as an open loop system as shown in Figure 2.3(a) and (b) then imposed with the constraint equations. After opening the closed loop, the open loop system is mapped using topology structure where the joint is numbered before it referred body as showed in Figure 2.3(c). This procedure is important when developing the recursive algorithm.



(a) A closed loop system

(b) Transformation into an open loop system by cutting the joint

(c) Topology of the open loop system

Figure 2.3: Cut joint approach for transforming a closed loop into an open loop system

From figure 2.3, B_0 is the ground body, while B_1, B_2, B_3 and B_4 are the number of bodies. The kinematics of the system is developed based on a recursive method in which the reference body is viewed as being located and oriented relatively to the preceding body. An example of the kinematics with the recursive method for a multibody system of two bodies interconnected by a joint is illustrated in Figure 2.4.

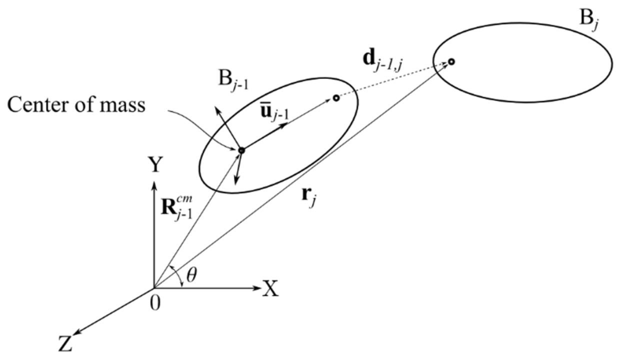

B$_j$
$\mathbf{d}_{j\text{-}1,j}$
B$_{j\text{-}1}$
Center of mass
$\bar{\mathbf{u}}_{j\text{-}1}$
Y
$\mathbf{R}^{cm}_{j\text{-}1}$
$\mathbf{r}_j$
θ
X
0
Z

Figure 2.4: Recursive body position description of multibody system

From figure 2.4, B$_j$ and B$_{j\text{-}1}$ represents the contiguous bodies which connect with a joint. The position of $\mathbf{r}_j$ can be defined in the global coordinate system as follows

$$\mathbf{r}_j = \mathbf{R}^{cm}_{j-1} + \mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1} + \mathbf{d}_{j-1,j} \tag{2.25}$$

where $\mathbf{R}^{cm}_{j-1}$ is the position vector of the center of mass of body B$_{j\text{-}1}$, $\mathbf{A}_{j\text{-}1}$ is the rotation matrix of the body B$_{j\text{-}1}$, $\bar{\mathbf{u}}_{j-1}$ is the position vector within body reference coordinate system and joint of body B$_{j\text{-}1}$, and $\mathbf{d}_{j-1,j}$ is the relative displacement vector between bodies. The velocity vector of joint, $\dot{\mathbf{r}}_j$ can be obtained by differentiating equation (2.25) with respect to time as

$$\dot{\mathbf{r}}_j = \dot{\mathbf{R}}^{cm}_{j-1} + \tilde{\boldsymbol{\omega}}_{j-1}\mathbf{u}_{j-1} + \dot{\mathbf{d}}_{j-1,j} \tag{2.26}$$

where $\dot{\mathbf{R}}^{cm}_{j-1}$ is the velocity of center of the mass of body B$_{j\text{-}1}$, $\tilde{\boldsymbol{\omega}}_{j-1}$ is the skew-symmetric matrix of body B$_{j\text{-}1}$ of the angular velocity, $\dot{\mathbf{d}}_{j-1,j}$ is the relative velocity between bodies and $\mathbf{u}_{j-1} = \mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1}$. The rotation matrix $\mathbf{A}_j$ and skew symmetric matrix $\tilde{\boldsymbol{\omega}}_j$ of the angular velocity of body B$_j$ can be written as:

$$\mathbf{A}_j = \mathbf{A}_{j-1}\mathbf{A}_{j-1,j} \tag{2.27}$$

$$\tilde{\boldsymbol{\omega}}_j = \tilde{\boldsymbol{\omega}}_{j-1} + \tilde{\boldsymbol{\omega}}_{j-1,j} \tag{2.28}$$

where $\mathbf{A}_{j\text{-}1,j}$ is the relative rotation matrix between two bodies and $\tilde{\boldsymbol{\omega}}_{j-1,j}$ is the skew symmetric matrix of the relative angular velocity between two bodies. The acceleration vector of joint $\ddot{\mathbf{r}}_j$ and relative angular velocity $\dot{\boldsymbol{\omega}}_j$ can be derived as:

$$\ddot{\mathbf{r}}_j = \ddot{\mathbf{R}}^{cm}_{j-1} + \dot{\tilde{\boldsymbol{\omega}}}_{j-1}\mathbf{u}_{j-1} + \tilde{\boldsymbol{\omega}}_{j-1}\tilde{\boldsymbol{\omega}}_{j-1}\mathbf{u}_{j-1} + \ddot{\mathbf{d}}_{j-1,j} \tag{2.29}$$

$$\dot{\boldsymbol{\omega}}_j = \dot{\boldsymbol{\omega}}_{j-1} + \dot{\boldsymbol{\omega}}_{j-1,j} \tag{2.30}$$

where $\ddot{\mathbf{R}}^{cm}_j$ is the acceleration of center of the mass, $\ddot{\mathbf{d}}_{j-1,j}$ is the relative angular acceleration between two bodies, $\dot{\tilde{\boldsymbol{\omega}}}_{j-1}$ is the first time derivative of the skew symmetric matrix of the relative angular velocity and $\dot{\boldsymbol{\omega}}_j$ is the angular acceleration of body B$_j$. From here, the derivation of the equations of motion can be formed using the Newton-Euler equations. By defining the acceleration of center of the mass and angular acceleration, the equations of motion can be written as:

$$\underbrace{\begin{bmatrix} m_j\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_j\mathbf{J}_j\mathbf{A}_j \end{bmatrix}}_{\mathbf{M}_j}\underbrace{\begin{bmatrix} \ddot{\mathbf{R}}^{cm}_j \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix}}_{\ddot{\mathbf{q}}_j} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \tilde{\boldsymbol{\omega}}_j\mathbf{J}_j\boldsymbol{\omega}_j \end{bmatrix}}_{\mathbf{Q}_{vj}} = \underbrace{\begin{bmatrix} \mathbf{F}_j \\ \mathbf{T}_j \end{bmatrix}}_{\mathbf{Q}_{ej}} \tag{2.31}$$

where $\mathbf{M}_j$ is the mass matrix, $\mathbf{J}_j$ is the inertia tensor, $\mathbf{F}_j$ is the external forces and $\mathbf{T}_j$ is the torsional forces of body B$_j$. Equations of motion for a multibody system using full set of generalized coordinates and constraint equations can lead to high computational costs if the system consists of a large number of connected bodies. In order to reduce the computational costs, the equations of motion for the system are presented with relative (joint) coordinates. This method uses a minimal set of generalized coordinates describing the equations of motion for the open loop system by implementing velocity transformation matrix. Therefore, the number of differential equations is reduced which can lead to increased computational efficiency [7] [24].

The constraints in the equations of motion can be taken into account using the velocity transformation matrix $\mathbf{V}$ and relative joint velocity $\dot{\mathbf{z}}$. The generalized velocity vector $\dot{\mathbf{q}}$ which contains the translational and angular velocities, can be projected with the minimum set of relative joint velocity $\dot{\mathbf{z}}$ using velocity transformation as

$$\dot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{z}} \tag{2.32}$$

where matrix $\mathbf{V} \in \mathbb{R}^{n_d \times n_f}$ is a velocity transformation matrix in which n_d is the number of generalized coordinates and n_f number of degrees of freedom. From equation (2.32), it can be seen that the matrix $\mathbf{V}$ play a major role in the formulation. To represents the generalized acceleration vector, equation (2.32) can be differentiated with respect to time as:

$$\ddot{\mathbf{q}} = \mathbf{V}\ddot{\mathbf{z}} + \dot{\mathbf{V}}\dot{\mathbf{z}} \tag{2.33}$$

where $\ddot{\mathbf{z}}$ is the relative joint accelerations and $\dot{\mathbf{V}}$ is the first time derivative of velocity transformation matrix. Adding equation (2.33) into the generalized equations of motion lead to

$$\mathbf{M}(\mathbf{V}\ddot{\mathbf{z}}+\dot{\mathbf{V}}\dot{\mathbf{z}})+\mathbf{Q}_{v}=\mathbf{Q}_{e} \tag{2.34}$$

To find the invertible form of the mass matrix, equation (2.34) has to be once again multiplied by the velocity transformation matrix $\mathbf{V}^{\mathrm{T}}$, which can be written with terms of the projected mass matrix $\mathbf{M}^{*}$ and force vector $\mathbf{Q}^{*}$ as follows:

$$\mathbf{M}^{*}\ddot{\mathbf{z}}=\mathbf{Q}^{*} \tag{2.35}$$

where $\mathbf{M}^{*}=\mathbf{V}^{\mathrm{T}}\mathbf{M}\mathbf{V}$ and $\mathbf{Q}^{*}=\mathbf{V}^{\mathrm{T}}(\mathbf{Q}_{e}-\mathbf{M}\dot{\mathbf{V}}\dot{\mathbf{z}}-\mathbf{Q}_{v})$. From the literature [4], the matrix $\dot{\mathbf{V}}$ does not need to be calculated explicitly. In fact, it is not computationally efficient to define it explicitly. Therefore, the coordinates $\ddot{\mathbf{z}}$ in equation (2.33) should be set to zero, which results in the relationship $\dot{\mathbf{V}}\dot{\mathbf{z}}=\ddot{\mathbf{q}}$. With this semi-recursive method, matrix **M*** and **Q*** can be computed in parallel, which improves computational efficiency. For the closed loop equations of motion, the penalty method suggested in equation (2.20) can be implemented as

$$(\mathbf{M}^{*}+\alpha\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{T}}\mathbf{\Phi}_{\mathbf{z}})\ddot{\mathbf{z}}=\mathbf{Q}^{*}-\alpha\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{T}}(\mathbf{\Phi}_{zt}\dot{\mathbf{z}}+\mathbf{\Phi}_{tt}+2\xi\omega_{n}\dot{\mathbf{\Phi}}+\omega_{n}^{2}\mathbf{\Phi}) \tag{2.36}$$

where $\mathbf{\Phi}_{z}$ is the Jacobian matrix of constraint with respect to the relative coordinates **z** and $\mathbf{\Phi}_{zt}$ is the first time derivative of Jacobian matrix of constraint. In the case of sceleronomic system, the time dependent of constraints can be eliminated from the equations. To reduce numerical problems due to the high penalty terms, the augmented Lagrangian method can be used.

Velocity transformation matrix

From equation (2.32), it is known the vector of relative joint velocity $\dot{\mathbf{z}}$ is equal to the number degrees of freedom of the system. This is reflect to the type of joint used in the system. Different type of joint may consist of single degree of freedom or more. As an example, two types of joints in the multibody system are illustrated in Figure 2.5.

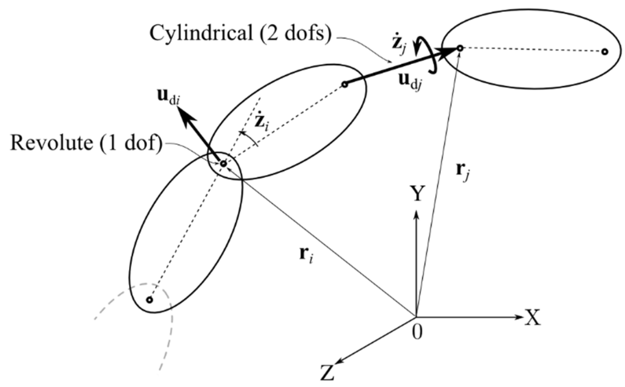


Figure 2.5: Expression of relative joint velocities, $\dot{\mathbf{z}}_i$ for revolute joint and $\dot{\mathbf{z}}_j$ for cylindrical joint

From Figure 2.5, revolute joint permit one degree of freedom where matrix $\mathbf{V}_{rev} \in \mathbb{R}^6$. From the literature [25], by defining vector $\mathbf{u}_{di}$ such that it is pointing the direction of the revolute joint, the velocity transformation matrix for revolute joint *i* can be written as:

$$\mathbf{V}_i^n = \begin{bmatrix} \mathbf{u}_{di} \times \left(\mathbf{R}_n^{cm} - \mathbf{r}_i\right) \\ \mathbf{u}_{di} \end{bmatrix}_{6\times1} \tag{2.37}$$

where $\mathbf{R}_n^{cm} - \mathbf{r}_i$ is the vector from revolute joint *i* to the center of mass of considered body *n*. In the case of cylindrical joint, which has two degrees of freedom, the velocity transformation matrix for cylindrical joint *j* can be written as:

$$\mathbf{V}_j^n = \begin{bmatrix} \mathbf{u}_{dj} \times (\mathbf{R}_n^{cm} - \mathbf{r}_j) & \mathbf{u}_{dj} \\ \mathbf{u}_{dj} & \mathbf{0} \end{bmatrix}_{6\times2} \tag{2.38}$$

where the matrix $\mathbf{V}_j^n \in \mathbb{R}^{6\times2}$ and $\mathbf{u}_{dj}$ is the vector pointing the direction of the cylindrical joint. Further detail explanations about the derivation of each type of joint can be found in [25].

In the open loop system, velocity transformation matrix $\mathbf{V}_i$ of each type of joint can be arrange easily based on it topology structure. This is explained graphically in Figure 2.6 with an assumption all joints are revolute joint.

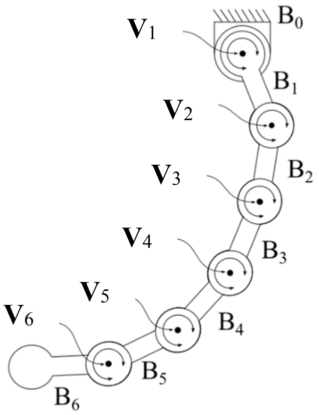


Figure 2.6: Description of an open loop system with revolute joint using velocity transformation matrix

To get the full transformation velocity matrix **V** for the assembled system in Figure 2.6, all dedicated matrices $\mathbf{V}_i^n$ need to be arranged into the level system matrix that represents all joints as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1^1 & & & \mathbf{0} \\ \mathbf{V}_1^2 & \mathbf{V}_2^2 & & \\ \vdots & .. & .. & \\ \mathbf{V}_1^6 & \mathbf{V}_2^6 & .. & \mathbf{V}_6^6 \end{bmatrix} \tag{2.39}$$

From equation (2.39), it can be concluded that the n^{th} row represents the effect of the n^{th} body through the other bodies to the base body in terms of relative velocities. For example, velocity transformation matrix which relate to body 3 can be written as

$$\mathbf{V}^3 = \left[\mathbf{V}_1^3\ \mathbf{V}_2^3\ \mathbf{V}_3^3\right] \tag{2.40}$$

where the subscript represents the corresponding joints related to considered body *n*. The transformation velocity matrix approach can provide computing efficiency as some of the matrices can be partially computed in parallel as proposed in [7]. The pattern of the matrices are also sparse, which can be beneficial if the sparse technique is implemented.

2.1.7 The flexible body description

A flexible body can be described using a floating frame of reference formulation. An example of a flexible body is given in Figure 2.7.

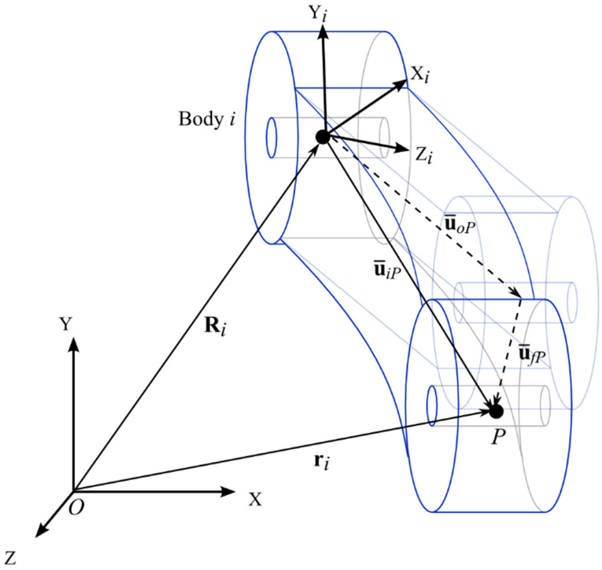


Figure 2.7: Coordinate system of flexible body deformed (blue line) and undeformed state (grey line)

As shown in Figure 2.7, the deformed position of point P with respect to the reference coordinate system is described using vector $\bar{\mathbf{u}}_{iP}$. As can be seen from figure, vector $\bar{\mathbf{u}}_{iP}$ consists of two components as follows:

$$\bar{\mathbf{u}}_{iP} = \bar{\mathbf{u}}_{oP} + \bar{\mathbf{u}}_{fP} \tag{2.41}$$

where vector $\bar{\mathbf{u}}_{oP}$ describes undeformed location of the point and vector $\bar{\mathbf{u}}_{fP}$ describes deformation of the point. Both vectors $\bar{\mathbf{u}}_{oP}$ and $\bar{\mathbf{u}}_{fP}$ are described with respect to reference coordinate system. The global position of point P of the flexible body i can be written as:

$$\mathbf{r}_{iP} = \mathbf{R}_i + \mathbf{A}_i(\bar{\mathbf{u}}_{oP} + \bar{\mathbf{u}}_{fP}) \tag{2.42}$$

To account for a number of discrete points within the flexible body i, vector $\bar{\mathbf{u}}_f$ need to be defined as follows:

$$\bar{\mathbf{u}}_f = \left[\bar{\mathbf{u}}_{f1}^{\mathrm{T}}, \bar{\mathbf{u}}_{f2}^{\mathrm{T}}, \ldots, \bar{\mathbf{u}}_{fn_p}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{2.43}$$

where n_p is the number of points used to describe a flexible body. In practical modeling, vector $\bar{\mathbf{u}}_f$ can be defined using assumed deformation modes obtained from a finite element model as:

$$\bar{\mathbf{u}}_f = \mathbf{\Phi}_d \mathbf{p} \tag{2.44}$$

where $\mathbf{p}$ is the vector of modal coordinates and $\mathbf{\Phi}_d$ is the matrix of assumed deformation modes which can be written as:

$$\mathbf{\Phi}_d = \left[\mathbf{\Omega}_1, \mathbf{\Omega}_2, ..., \mathbf{\Omega}_{n_m} \right] \tag{2.45}$$

where $\mathbf{\Omega}_i$ is an assumed deformation mode defining translational coordinates of discrete point and n_m is the number of modes. The matrix of assumed deformation modes $\mathbf{\Phi}_d$ can contain the same number modes as the degrees of freedom in the original finite element model. Accounting for all deformation modes, computational cost maybe overly high. In practical modeling, the number of deformation modes can often be considerably smaller than the number of degrees of freedom in the original finite element model without significant loss of accuracy. Accordingly, a model order reduction method can be used to reduce the size of matrix $\mathbf{\Phi}_d$.

In this dissertation work, the Guyan reduction method (leading to matrix $\mathbf{\Phi}_g$), the Craig-Bampton method (leading to matrix $\mathbf{\Phi}_{cb}$), and the Krylov subspace methods (leading to matrix $\mathbf{\Phi}_k$) are all implemented to obtain the matrix of assumed deformation modes $\mathbf{\Phi}_d$. These methods make it possible to reduce the set of equations of motion with respect to the flexible bodies, and therefore, they lead to decreased numerical solution time. This makes the finite element model suitable for the multibody application. The development of the equations of motion for flexible bodies can be based on the same procedures described previously, but the deformation modes must be considered when deriving the equations of motion as described in [26].

Model order reduction

Guyan reduction method partitions the stiffness matrix $\mathbf{K}$ of the original finite element model into submatrices (boundary, B and internal, I) and makes the approximation that there is no external load applied on the internal degrees of freedom [27]. The technique reduces the finite element model by condensing the internal degrees of freedom [28]. Using this method, assumed deformation modes can be described with help of the static reduction projection matrix $\mathbf{\Phi}_g$ as follows:

$$\mathbf{\Phi}_g = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} \end{bmatrix} \tag{2.46}$$

The eigenvalues obtained from the reduced system have a tendency of being higher compared to the original system, which means that the degrees of freedom selection process is critical to the success of Guyan reduction.

The Craig Bampton method is motivated by the need to model the subcomponents of large structures separately. To reduce the dynamic model of each subcomponent, the degrees of freedom at the boundaries which join the subcomponents must be retained [29]. The internal dynamic of each subcomponent can be represented by an eigenmode with fixed boundaries. In Craig-Bampton method, two types of modes need to be considered which are fixed boundary modes, $\mathbf{\Phi}_I$ and static modes, $\mathbf{\Phi}_B$ [30]. These two set of modes can be combined as follows:

$$\mathbf{\Phi}_{cb} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{\Phi}_B & \mathbf{\Phi}_I \end{bmatrix} \tag{2.47}$$

Craig-Bampton modes $\mathbf{\Phi}_{cb}$ can be modified with help of orthonormalization procedure in which equation (2.47) used to reduce the size of original mass and stiffness matrices. The eigenvalue analysis is then performed to reduced matrices to obtain orthogonal deformation modes.

During last years, Krylov subspace methods have become favourable for finding reduced model. Krylov subspace methods are iterative methods which provide an approximation solution of large-scale linear systems and eigenvalue problems [31]. Krylov subspace methods can make a reduction from dimension m_o to dimension m_r where $m_r << m_o$ based on the spaces spanned by successive groups of sequence m_r column vectors as:

$$\mathbf{\Phi}_{k,m_r} = \text{span}\left\{\mathbf{b}\ ,\mathbf{W}\,\mathbf{b}\ ,\mathbf{W}^2\mathbf{b}\ ,\ldots,\mathbf{W}^{m_r-1}\mathbf{b}\ \right\} \tag{2.48}$$

For the case of the second order undamped system [32], $\mathbf{W} = \mathbf{K}^{-1}\mathbf{M}$ and $\mathbf{b} = \mathbf{K}^{-1}\mathbf{B}_\mathrm{c}$ where $\mathbf{B}_\mathrm{c}$ is the constant force vectors and the selected span is called the Krylov projection matrix, $\mathbf{\Phi}_{k,m_r} = \mathbb{R}^{m_o \times m_r}$. The original equation is reduced by multiplication with the Krylov projection matrix which reduces the dimensions of the system matrices of mass and stiffness [33] [34].

There are two popular iteration methods that are commonly used in the Krylov subspace algorithm: Arnoldi and Lanczos iterations [35]. The Lanczos and Arnoldi methods are functional for transforming a Hermitian matrix to tridiagonal form and a non-Hermitian matrix to Hessenberg form. In the literature [36], it is pointed out that Krylov subspace based methods are robust and computationally efficient for large matrices with small reduction.

2.2 Hydraulic systems modeling

Hydraulic systems can be modeled using lumped volume theory [18]. When this theory is applied, there is an assumption that the effect of the acoustic wave is neglected. In the lumped volume theory approach, the hydraulic circuit is divided into volumes within which the pressure is assumed to be equally distributed. Differential equations are formed for the volumes to obtain the pressure of the system. Different volumes into which the fluid can flow are separated by throttles. The flow rate can be determined based on pressure difference. Hydraulic systems normally have high nominal frequencies. Therefore, in order to obtain good results the time step must be short. A simple hydraulic system consisting of a directional valve, restrictor valve and cylinder with implementation of lumped fluid theory is shown in Figure 2.8.

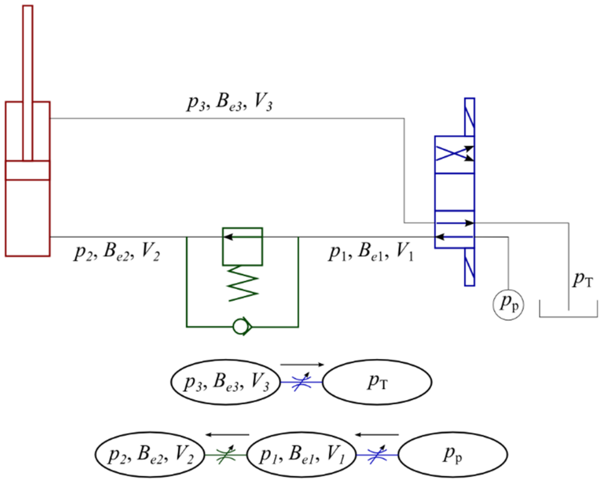


Figure 2.8: Simple hydraulic system and its lumped fluid theory representation

In Figure 2.8, p_i is the pressure, B_{ei} is the effective bulk modulus, V_i is the volume size at volume i and p_P and p_T are the pressure at the pump and tank, respectively. The differential equation for hydraulic pressure at each volume i can be defined as:

$$\dot{p}_i = \frac{B_{ei}}{V_i}\left(Q_{in,i} - Q_{out,i} - \dot{V}_i\right) \tag{2.49}$$

where $Q_{in,i}$ and $Q_{out,i}$ is the incoming and outgoing flow rates at volume i, and $\dot{V}_i$ is the change of volume V_i with respect to time, respectively. In equation (2.49), B_{ei} is the effective bulk modulus at volume i. The effective bulk modulus accounts the bulk modulus of the fluid, the flexibility of container, and dissolved air [37].

The volume flow Q_t in a restrictor valve can be written as:

$$Q_t = C_d A_t \sqrt{\frac{2(p_{in} - p_{out})}{\rho}} \tag{2.50}$$

where C_d is the discharge coefficient, A_t is the cross-section area of the valve, ρ is the density of the fluid, p_{in} is the input pressure and p_{out} is the output pressure. In this study, the volume flow is described using semi-empirical methods in which the parameters of the valve are obtained from a manufacturer's catalogue.

A hydraulic cylinder can be modeled based on its dimensions and the input pressure, which can be obtained from equation (2.49). The model of cylinder can be formed based on a free body diagram as shown in Figure 2.9.

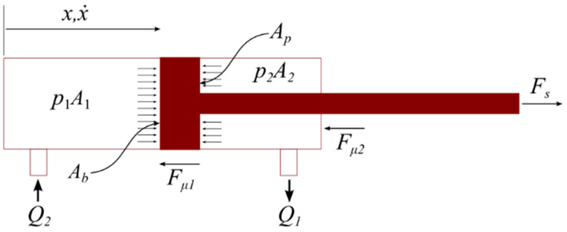


Figure 2.9: Forces acting on a hydraulic cylinder

The hydraulic cylinder volume is solved with the function of the cylinder stroke as:

$$\begin{aligned} V_1 &= xA_b \\ V_2 &= (l-x)A_p \end{aligned} \tag{2.51}$$

where A_b and A_p are the area behind the piston side and the area of the cylinder piston rod side, respectively, x is piston displacement and l is the maximum stroke length. The flow rate Q produced base on the piston motion can be written as:

$$\begin{aligned} Q_1 &= -\dot{x}A_b \\ Q_2 &= \dot{x}A_p \end{aligned} \tag{2.52}$$

where $\dot{x}$ is the piston displacement velocity. The force F_s produced by the cylinder can be written as:

$$F_s = p_1A_1 - p_2A_2 - F_\mu \tag{2.53}$$

where $F_{\mu} = F_{\mu 1} + F_{\mu 2}$ is the total friction force, and p_1 and p_2 are the pressure in the cylinder chambers. The friction force can be calculated as a function of cylinder efficiency η and velocity as:

$$F_{\mu} = (p_1 A_1 - p_2 A_2)(1-\eta) f(\dot{x}) \tag{2.54}$$

where $f(\dot{x})$ is the velocity dependent coefficient, which can be obtained based on the speed of the average measurement from different cylinders [38].

2.3 Collision and contact modeling

Contact modeling is an important modeling element when analyzing real multibody systems. It is functional to determine the collision point and provide the response of the collision. In the working mobile vehicle simulation, contact model is needed to determine the contact force between two bodies.

There are two main steps in modeling a contact: collision detection and collision response. The collision detection model determines the location and time of the collision while collision response is to provide the contact force between the two bodies. As the procedures need to be done within the real time period, proper contact algorithm need to be used to determine when and where contact occurs, and to calculate the response of the contact.

A collision detection between bodies with different geometry can be defined using bounding volume (BV). This method is effective for approximating whether objects are overlapping. BV uses spheres or boxes, which condense the complex geometrical object into a simple shape. Using these simple shapes, the collision detection process can be accelerated. A number of different bounding methods can be used for collision detection, such as Axis-Aligned Bounding Box (AABB), and Oriented Bounding Box (OBB) which described in [39] [40].

Several methods can be used in in collision response, such as the penalty methods, analytical methods and impulse methods [41] [42] [43]. In contact modeling, penalty methods are the procedure which allow small penetrations between two bodies at the contact point which this method can be called as soft contact. The penetration distance is added with the spring-damper elements. The contact force is computed and applied as an external force into the equations of motion. In the analytical methods, the constraints are taken into consideration to solve the contact. However, this method will increase the dimension of the equations of motion and solving all contacts simultaneously may lead to computational cost. In the impulse methods, the collision between two bodies occur without solving the contact force. This method will compute the velocities at the contact point and applied the new velocities directly to the bodies.

When a collision occurs, the current simulation stopped and a new initial condition will be used in the next time step of simulation. It should be noted that in order to obtain correct results, correct time-step value selection is crucial, especially when dealing with bodies moving at high velocity. Small time step may increase the computing time and large time step may allow the overly large penetration between two bodies. However, it is often difficult to avoid penetration between the objects, despite the use of smaller time-steps in the collision detection algorithm.

In this study, the bounding box is applied for the collision detection and the penalty method for the collision response. Figure 2.10 shows the general algorithm for contact between two bodies using penalty method.

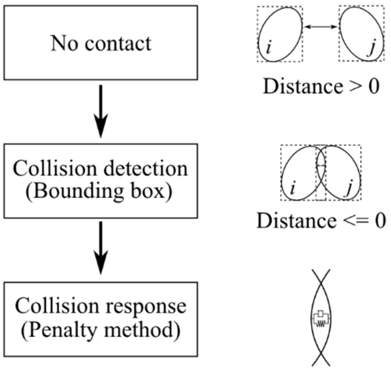


Figure 2.10: General algorithm for contact between two bodies

The penetration can give inaccurate simulation and produce errors in the simulation system. In general, the kinematic of the contact point can be described as in Figure 2.11.

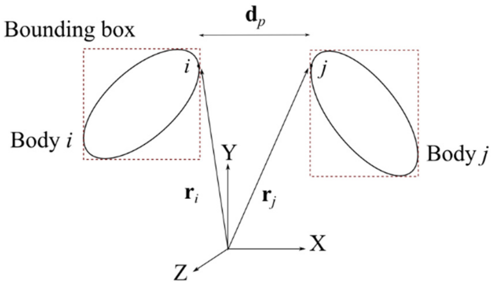


Figure 2.11: Contact between two bodies using bounding box approach

From figure 2.11, the position of the two points can be derived using multibody formulations. The distance, $\mathbf{d}_p$ between two points i and j can be derived as:

$$\mathbf{d}_p = \mathbf{r}_j - \mathbf{r}_i \tag{2.55}$$

The normal vector **n** of the contact can be determined as:

$$\mathbf{n} = \frac{\mathbf{d}_p}{\left\|\mathbf{d}_p\right\|} \tag{2.56}$$

The normal magnitude between two points d_p can be obtained as:

$$d_p = \mathbf{n}^{\mathrm{T}}\mathbf{d}_p \tag{2.57}$$

The relative normal velocity v_n between two points can be obtain by differentiating equation (2.55) with respect to time as:

$$v_n = \mathbf{n}^{\mathrm{T}}(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i) \tag{2.58}$$

In the case of collision, where d_p becomes as penetration distance at contact point, a spring-damper is added at the contact point to describe the contact forces. The normal contact force, $\mathbf{F}_n$ at the contact point can be formulated as:

$$\mathbf{F}_n = -(Kd_p + Cv_n)\mathbf{n} \tag{2.59}$$

where K and C are the coefficients of stiffness and the damping factor. The values for the two coefficients K and C is vital and may result in a range of desired collision response types. The penalty method can nearly fulfill the exact contact conditions, although its accuracy depends strongly on the penalty parameters and the specific case.

2.4 Friction modeling – The tire model

The tire transmits forces between the road and the rim, and thus friction description between the tire and the ground is critical when a tire is modelled. A realistic model can be obtained by considering the contact patch interaction between the tires and ground using finite element methods. The detailed finite element models needed in the contact patch description are, however, associated with high computing cost, which creates difficulties when results should be obtained in real time. Therefore, simplified tire models must be used. In this study, the lumped LuGre model has been chosen [17].

The LuGre tire model used in this work is derived based on description of an elastic bristle at the microscopic level. At the contact point, the tangential force causes the bristles to deflect similarly to springs, and creates the friction force. When a large tangential force is imposed, the bristles first deflect and then start to slip. An overview of the concept underlying the LuGre model can be seen from Figure 2.12(a) and (b).

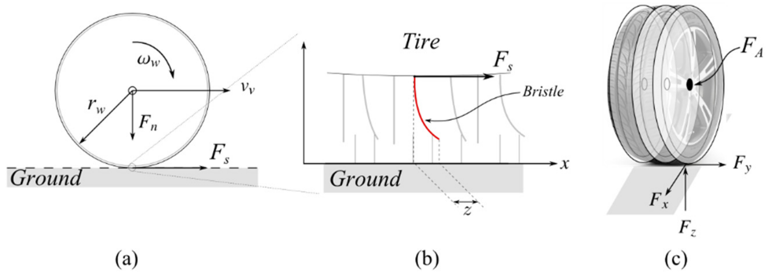


Figure 2.12: LuGre's tire friction model

The LuGre friction model is an enhancement of the Dahl model in which the Stribeck effect taken into account [44]. The enhancements bring many benefits because the LuGre friction model is able to take account of many friction effects such as stiction, the Stribeck effect, stick slip, zero slip displacement and hysteresis. The bristle deflects with respect to time when the tire rotates and the deflection of the bristle, represented by z, can be written as:

$$\frac{dz}{dt} = v_r - \frac{\sigma_0 |v_r|}{g_f} z \tag{2.60}$$

where g_f is friction, σ_0 is the rubber longitudinal lumped stiffness and v_r is the relative velocity between two sliding surfaces defined as:

$$v_r = r_w \omega_w - v_v \tag{2.61}$$

where r_w is the tire radius, ω_w is the tire angular velocity, v_v is the vehicle velocity, and friction g_f can be represented as:

$$g_f = \mu_c + (\mu_s - \mu_c) e^{-\left|\frac{v_r}{v_s}\right|^{-\frac{1}{2}}} \tag{2.62}$$

where μ_c is normalized Coulomb friction, μ_s normalized static friction, and v_s is the Stribeck relative velocity. Friction, g_f must always be positive, and it depends on the material properties, temperature, and other factors. Friction force based on the bristle deformation can be expressed as follows:

$$F_s = (\sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 v_r)F_n \tag{2.63}$$

where σ_1 is the longitudinal lumped damping coefficient, σ_2 is viscous relative damping, and F_n is the normal force.

To obtain a computationally efficient and accurate tire model, a tire can be presumed to be a series of discs, as shown in Figure 2.11(c), based on the width of the tire. In this approach, the disks are assumed to be simple rigid bodies, and the profile of the ground can be constructed based on the required environment (off-road profile). In the LuGre tire implementation, all forces are applied to the hub of the tire, denoted as F_A.

3 Sparsity in multibody methods

This section will discuss sparsity in equations of motion for multibody systems. When dealing with the numerical programming, computational efficiency is always the main goal and need to give attention. The multibody formulations use the matrices to solve the systems of equation. It has been found that several matrices playing a major role in the equations of motion are large and sparse. Therefore, to solve system of matrices efficiently, the sparse matrix technique need to be employed. Firstly, this technique will only account the nonzero entries to be saved in the computer memory [45]. As only the nonzero entries is considered, huge amount of computer memory can be saved and may increase the computational efficiency. Secondly, the sparse linear systems can be solved using direct method or iterative method. The most common direct method are Cholesky factorization and LU decomposition while for iterative method is Conjugate Gradient [46] [47].

3.1 Sparsity in the equations of motion

For any case, solving the linear equations of motion involves constructing three main vector-matrix matrices: matrix **W**, vector **x** and vector **b**. For example, an equations of motion with Lagrange multipliers in matrix form is constructed as:

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{Q}_e - \mathbf{Q}_v \\ \mathbf{Q}_c \end{bmatrix}}_{\mathbf{b}} \tag{3.1}$$

When solving $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ in real-time simulation, the sparsity of the matrix **W** needs to be considered to efficiently solve the equation. In this equation, there are two matrices that potentially have sparse properties: the mass matrix, **M**, and the Jacobian matrix of constraints, $\mathbf{\Phi}_{\mathbf{q}}$. The dimension of matrix **M** becomes more radical for a spatial model as it will increase in dimension compared to a planar model. In the spatial model, the mass matrix is arranged diagonally, and the total mass matrix with *n* number of bodies can be written as:

$$\mathbf{M}_i = \begin{bmatrix} m_i\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix}_{6\times 6}, \quad \mathbf{M} = \mathrm{diag}\left[\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\right]_{6n\times 6n} \tag{3.2}$$

From equation (3.2), the sparsity of the mass matrix can be described as shown in Figure 3.1. The number of zero entry increases exponentially with respect to the number of bodies.

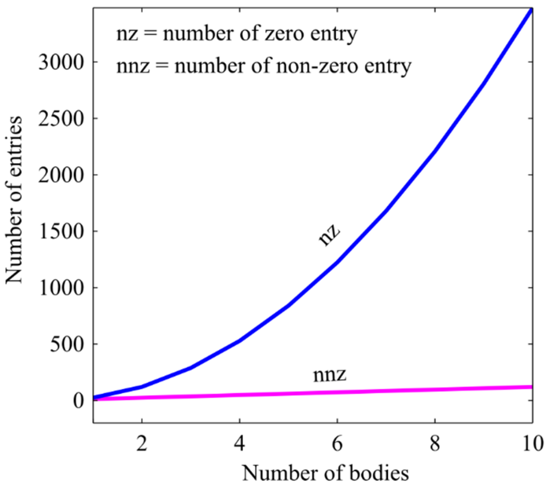


Figure 3.1: Sparsity description for the mass matrix, **M**

From Figure 3.1, it can be seen that the number of non-zero elements in mass matrix **M** increases dramatically with the number of bodies. In the case under consideration, the matrix is symmetric and positive definite. Another matrix that can have sparsity is the Jacobian matrix of constraints, $\mathbf{\Phi_q} = \mathbb{R}^{n_c \times n_g}$, where n_c is the number of constraint equations, and n_g is the number of generalized coordinates. For example, if the multibody system have a multiple joints in the spatial system, constraint equations of all joints need to be derived. The constraint equation of joint is formulated based on the two bodies connected at the joint. If the joint is a revolute joint, five constraint equations need to be added into the vector of constraint $\mathbf{\Phi}$. If the system has n_r number of revolute joints, the number of constraint equations increases to $5n_r$. Therefore, it can be said the Jacobian matrix $\mathbf{\Phi_q}$ in a large scale constrained system will become sparse and not symmetric. When the mass matrix **M** is coupled with the Jacobian matrix $\mathbf{\Phi_q}$ and its transposition is arranged into matrix form as in equation (3.1), the augmented matrix **W** becomes very sparse and symmetric.

In the case of the velocity transformation matrix, the sparsity of the velocity transformation matrix **V** depends on the type of joint, number of joints and the arrangement of the bodies. If the topology structure is a tree structure, the sparsity increases with the number of branches. Due to the assemble matrix **V** is based on the topology structure, the matrix **V** can be lower or upper triangular.

Therefore, it can be concluded that the dimension and sparsity in the system matrices of the equations of motion increase rapidly with the number of bodies, even at small numbers of degrees of freedom. The mass matrix **M** and the Jacobian matrix $\mathbf{\Phi_q}$ play a significant role in the equation and the sparsity of these matrices has a considerable effect on the computing performance [48].

3.2 Sparse solution methods

To increase the computational efficiency of solving the system of linear equations **Wx**=**b**, a sparse technique approach with suitable methods needs to be implemented. Based on knowledge of the type of matrix, appropriate selection of the method used to solve the linear equation can shorten the computing time such that it becomes to real time or near to it. In most cases of real-time multibody simulation, the matrix **W** is symmetric and positive definite, and thus $\mathbf{W} = \mathbf{W}^{\mathrm{T}}$ and $\mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{x} > 0$ where $\mathbf{x} \in \mathbb{R}^{n}$. In such cases, memory requirements can be reduced , which reduces CPU time, as only elements at the upper or lower triangle need to be stored in the memory. There are two approaches to solving a sparse system for a linear equation: a direct method or an iterative method [49]. The most common solver used in the direct method is Cholesky decomposition, and for the iterative approach the Conjugate Gradient method.

In Cholesky factorization decomposition, a symmetric and positive-definite matrix can be factorized into the product of a lower triangular matrix and its transpose, where $\mathbf{W} = \mathbf{C}\mathbf{C}^{\mathrm{T}}$ and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is called the Cholesky factor. An example is given in Figure 3.2 where the matrix is a lower triangular with positive diagonal entries. Therefore, the triangular system can be used to solve $\mathbf{C}\mathbf{y} = \mathbf{b}$ for **y** and then $\mathbf{C}^{\mathrm{T}}\mathbf{x} = \mathbf{y}$ for **x**.

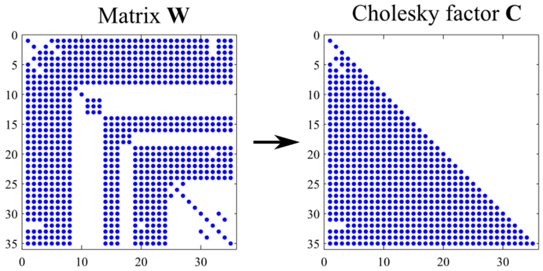


Figure 3.2: Transformation of matrix **W** to Cholesky's factor **C** matrix

The linear equation also can be solved using an iterative approach such as the Conjugate Gradient (CG) method. CG is an algorithm to obtain a good approximation of the solution $\mathbf{x}^{*}$. CG starts with an initial guess solution as $\mathbf{x}_0$ and then searches for the solution at each iteration until the solution is close to $\mathbf{x}^{*}$. The CG algorithm starts by generating vector sequences of iterates, residuals corresponding to the iterates, and search directions used in updating the iterates and residuals. Even though the length of this sequence possibly becomes large, only a small number of vectors have to be kept in the memory. An advantage of this method is the simplicity of the algorithm to determine the new direction vector. Thus, the CG method is suitable for a large system for which direct method may take longer time to solve [50]. This operation can be done very efficiently by a properly stored sparse matrix.

4 Programming libraries and numerical examples

Different sub-models need to be executed in real-time working mobile vehicle simulation. These models include multibody modeling, hydraulic modeling, and contact and friction modeling. All of the modeling must be done simultaneously within the real-time period. In this section, the multibody modeling methods will be concentrated to show the difference of computational efficiency.

The multibody system is solved using an augmented Lagrangian method and a semi-recursive method using velocity transformation matrix. The explicit Runge-Kutta with fixed time step is used to solve the equations. Full and sparse matrix techniques are implemented in the development code. The solver for the real-time simulation is coded in ANSI C in order to fully utilize the portability and flexibility with different operating systems and computer architecture.

The results is obtained from MeVEA dynamic solver module where it is responsible for the mechanics and hydraulic simulation [51]. In order to ensure the results is accurate, reliable and robust, sparse libraries are in used. There are several reliable and well known free licences (GPL) solver available in the market which easily to be integrated with the C code can be used for solving the linear equations for dense and sparse format such as LAPACK and UMFPACK.

4.1 Programming libraries

When the matrix is sparse, the efficiency of the arithmetic operation can be increased by manipulating the matrix structure (factorization etc.) and handling only the nonzero elements. This approach can save memory allocation during the computational period and can considerably increase computational efficiency.

There are several ways to efficiently store a sparse matrix. Each technique has its own advantages in terms of the storage scheme, but the suitability of the techniques depends on the processor architecture and libraries solver. The most common sparse formats are the coordinate format (COO), compressed sparse row format (CSR), compressed sparse column format (CSC) and Harwell Boeing sparse format (HB) [52]. The compressed sparse column format and Harwell Boeing sparse format have been chosen for use in this study as it is the format use by the commercial software such as Ansys and easy to integrate with the available numerical solvers.

Full matrix

$$\begin{bmatrix} 2 & 0 & 0 & 10 & 0 & 12 \\ 0 & 2 & 0 & 0 & 6 & 2 \\ 0 & 0 & 55 & 77 & 0 & 98 \\ 23 & 0 & 21 & 32 & 0 & 2 \end{bmatrix}$$

Compressed Sparse Column

Col_Ptr	1		3	4		6			9				10	14
Row_Ind	1	4	2	3	4	1	3	4	2	1	2	3	4	
Element	2	23	2	55	21	10	77	32	6	12	2	98	2	

Figure 4.1: Compressed sparse column format

Figure 4.1 shows the transformation from full matrix format into CSC format. CSC provide three items of information: row index (Row_Ind), to store row indices of non zero elements; column pointer (Col_Ptr), to store the index of the starting element in the column of the matrix; and Element is the list of non-zero elements.

The Harwell-Boeing (HB) format provides useful information about the properties of the sparse matrix, such as matrix type, number or row and column, number of data lines, number of non-zero elements, and matrix character, such as unsymmetrical, Hermitian and many others. HB uses the CSC format to represent the location of the nonzero elements and it values. This format is the format of the finite element software used in this study to obtain the stiffness and mass matrices in order to run the model order reduction procedures.

Several numerical libraries are freely available that can be used to solve the linear equation in either full or sparse matrix format. Figure 4.2 shows libraries that have been used and tested for verification in the development code.

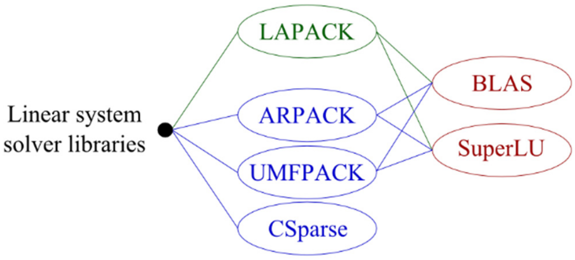


Figure 4.2: Free solvers available for use in simulation

All of these solvers are free software under the terms of the GNU general public license. Most powerful numerical software, such as GNU Octave, SciPy and Matlab, uses these libraries as the backbone of their linear equation solver. All of these libraries (except CSparse) are developed based on Fortran language, which is a powerful language for scientific programming. LAPACK is a solver for simultaneous linear equations, eigenvalue problems, least-squares solution and the singular value problem [53]. It was specifically developed to handle both dense and banded matrix formats. UMFPACK is functional for solving linear equations in which the matrix **W** is sparse and not required to be symmetric. The matrix can be rectangular or square and it can also be singular. The UMFPACK solver factorizes the matrix into the product LU, which are the lower and upper triangular of the matrix **W**, before solving the equation [54]. CSparse is a simplification routine in UMFPACK specifically developed in C to solve sparse linear equations when matrix **W** is symmetric or not [49]. ARPACK is a solver for really large scale eigenvalues problems. It is developed based on the implicitly restarted Arnoldi iteration method [55]. This solver can be used to solve symmetric or unsymmetrical problems to obtain a few eigenvalues and eigenvectors, either the largest or the smallest. It is also offers some routines for solving banded matrices. It is claimed that this solver is portable, robust and efficient for solving eigenvalue and eigenvectors with large matrices. The solver uses a reverse communication approach in which the matrix can be stored in many ways and a direct matrix-vector product is not required. ARPACK is used when dealing with the flexible multibody procedure in which the mode shape is required to measure the body deformation.

In order to perform efficiently, LAPACK, UMFPACK and ARPACK need to be associated with support solvers, which are BLAS and SuperLU routines. BLAS is a basic linear algebra subprogram that provides efficient and portable basic arithmetic solutions for vector and matrix operations [56]. SuperLU is used to solve large and sparse matrices with nonsymmetric linear equations using a direct solution [57]. All the libraries mentioned use the same sparse format, CSC, for synchronization.

4.2 Numerical examples

To determine the computing performance of the different methods, two mobile working vehicles, a Logset H8 tree harvester and Normet concrete spray (NCSS), shown in Figure 4.3, were modeled using the augmented Lagrangian method and the semi-recursive method. The mobile working vehicles have a different number of degrees of freedom, 27 for Logset H8 and 10 for NCSS, respectively.

(a) Logset tree harvester

(b) Normet concrete spray

Figure 4.3: Working mobile vehicles for which the models were implemented (adapted with permission from Mevea Ltd.)

With the semi-recursive method, the topology of the machines needs to be developed, as shown in Figure 4.4 for the Logset H8 tree harvester, in order to derive the equations of motion. To map the topology structure, the closed loop needs to be opened via a cutting joints procedure, and the system can then to be treated as an open loop system. The opened joints are then imposed with a penalty formulation to obtain the dynamic equilibrium.

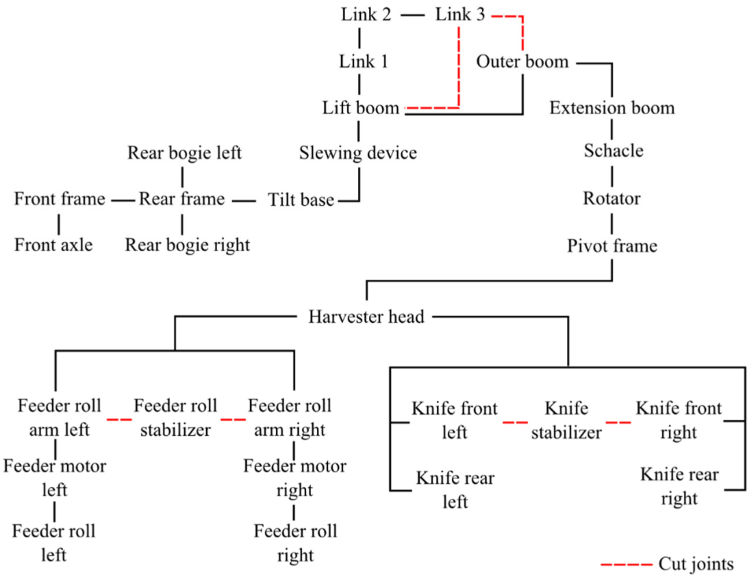


Figure 4.4: Detailed topology of the Logset H8 tree harvester

The velocity transformation matrix **V** is constructed from the topology structure. From the equations of motion for the semi-recursive method, in order to solve $\ddot{\mathbf{q}}$, mass matrix **M** must be multiplied with the velocity transformation matrix **V**. As illustrated in Figure 4.5, the actual matrix **V** and mass matrix **M** are highly sparse, and the sparse matrix technique can thus be implemented.

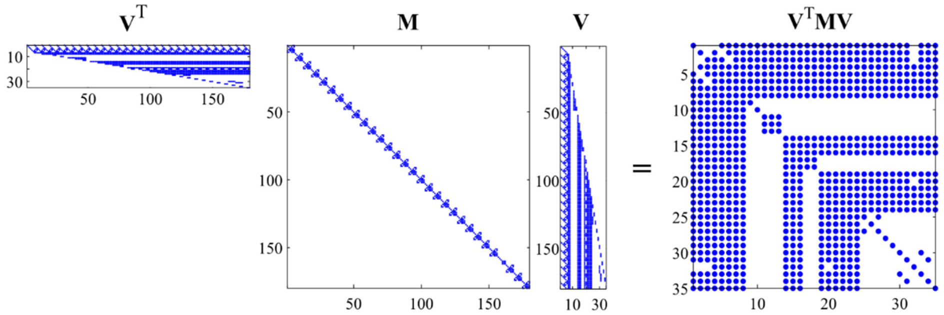


Figure 4.5: Sparsity in matrix **V** and **M** for the Logset H8 tree harvester

From Figure 4.5, by using sparse approach the matrix of $\mathbf{V}^T\mathbf{M}\mathbf{V}$ can be implemented with the Cholesky factorization to solve the linear equation system as in equation (2.36). As a comparison, the sparsity of the matrices in the equations of motion for both models was recorded and is given, together with the number of degrees of freedom and other related information, in Table 4.1.

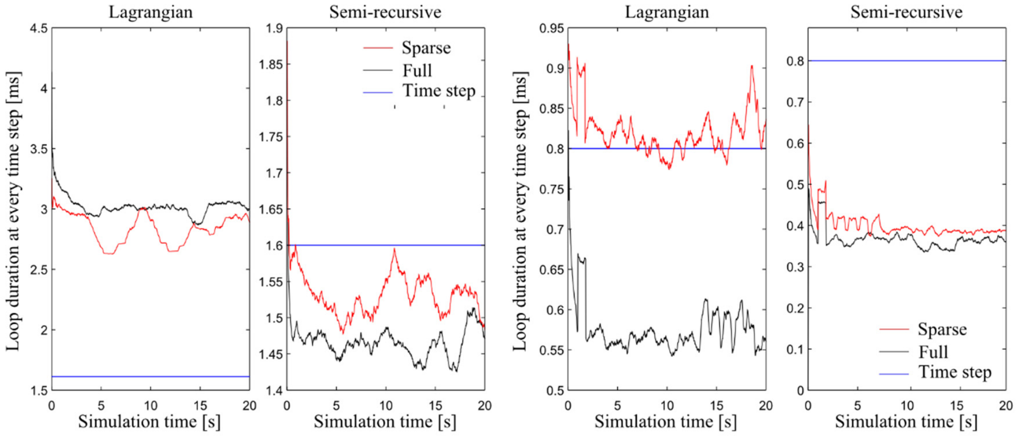
Table 4.1: Sparsity description data

Description	Normet concrete spray		H8 Tree Harvester	
	Lagrangian	Recursive	Lagrangian	Recursive
Number of bodies	11	11	30	30
Generalized coordinates	66	66	180	180
Joint coordinates	-	12	-	35
Constraint equations	56	2	153	8
Number of DOFs	10	10	27	27
Number of entries of $\boldsymbol{\Phi}_{\mathbf{q}}$, $n_{\boldsymbol{\Phi}\mathbf{q}}$	3696	132	27540	1440
Number of non-zero entries, $nnz_{\boldsymbol{\Phi}\mathbf{q}}$	432	24	1128	78
$nnz_{\boldsymbol{\Phi}\mathbf{q}}$ / $n_{\boldsymbol{\Phi}\mathbf{q}}$ (%)	11.69_%	18.18_%	4.1_%	5.42_%
Number of entries of **M**, $n_{\mathbf{M}}$	4356	4356	32400	32400
Number of non-zero entries, $nnz_{\mathbf{M}}$	330	330	900	900
nnz_M/n_M (%)	7.58_%	7.58_%	2.78_%	2.78_%
Number of entries of **V**, n_V	-	792	-	6300
Number of non-zero entries, nnz_V	-	66	-	450
$nnz_{\mathbf{R}}/n_{\mathbf{R}}$ (%)	-	8.33%	-	7.14_%

It is clear from Table 4.1 that sparsity occurs in the system of matrices. The computational efficiency for the augmented Lagrangian and semi-recursive methods with and without the sparse matrix technique are described in Figure 4.6 for the NCSS and Logset H8 models, respectively. The computation loop duration was recorded for every solution for 20 seconds and synchronized with the real clock time in order to show the performance of each method. The loop duration was then compared with the fixed time step of the fourth order Runge-Kutta method (0.8ms for NCSS and 1.6ms for Logset H8).

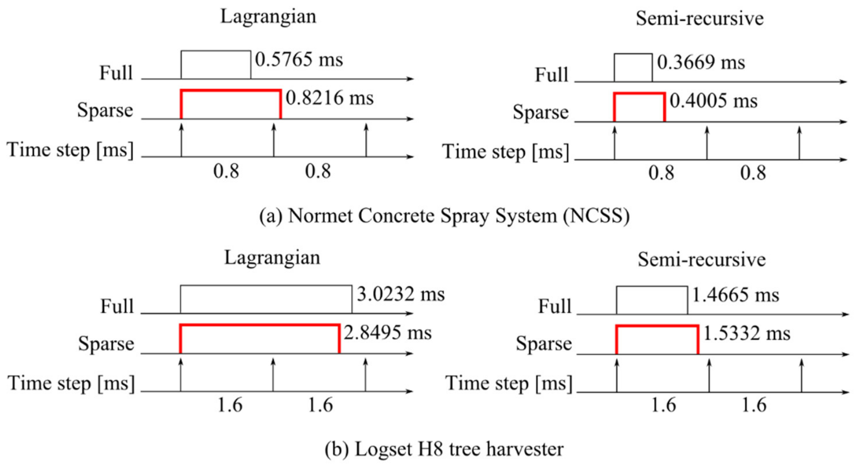
For the Logset H8 tree harvester, the augmented Lagrangian method required more time to solve the linear system at each loop, and when the sparse technique was employed, the computational time did not show any significant improvement. However, when the semi-recursive method is employed, the time to solve the linear system decrease almost two times than the augmented Lagrangian method. Same as in the augmented Lagrangian method, when the matrices are executed with the sparse technique, the computational time at each loop slightly increases, although it is still less than the reference time step.

For the NCSS solution, the computational time at each loop reduced significantly when the semi-recursive method was applied compared with the augmented Lagrangian method. The sparse technique slightly increase the computing time with both the augmented Lagrangian and semi-recursive methods. From the results, the semi-recursive method with the full matrices approach is computationally more efficient than the augmented Lagrangian method. This is due to the small number of degrees of freedom, which leads to a small size of the matrices.



(a) Logset H8 tree harvester (b) Normet Concrete Spray System (NCSS)

Figure 4.6: Comparison of numerical simulation results between Lagrangian and semi-recursive methods for Logset H8 and NCSS during 20 seconds simulation



(a) Normet Concrete Spray System (NCSS)

(b) Logset H8 tree harvester

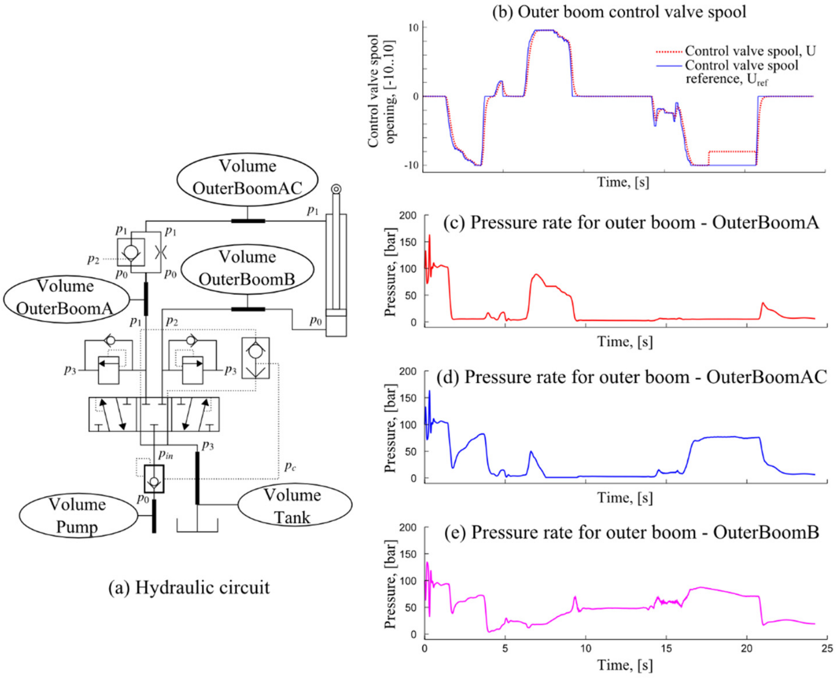
Figure 4.7: Average equation of motion - solver loop duration for the NCSS and the Logset H8 models versus time step

Figure 4.7 shows the average loop duration for the two methods, both with and without the sparse technique, and compares them against the fixed time step. For the NCSS model, the usage of the sparse implementation technique increases the computational time for both the augmented Lagrangian method and the semi-recursive method, and it reduces the computing efficiency to 42.52% and 9.16%, correspondingly, when compared with full matrix implementation. For the Logset H8 model, the sparse implementation technique in the augmented Lagrangian method helps to reduce the computing time by 5.75% when compared with full matrix implementation. In the case of the semi-recursive method, the sparse implementation technique increases the computing time by 4.55% compared with full matrix implementation.

From observation of the simulation performance for both methods, augmented Lagrangian and semi-recursive methods worked as expected for both working mobile vehicles models. Implementation of the semi-recursive method significantly improved the computational efficiency compared with the augmented Lagrangian method. However, when the sparse technique is implemented into both methods, only augmented Lagrangian method for Logset H8 tree harvester shows slightly improvement in term of computational efficiency.

It can be concluded from the results that sparse matrix technique can offer a computational efficiency improvement for the augmented Lagrangian method if the system of matrices in the equations of motion are large. This is because in this method, the dimension of the matrices are based on the number of generalized coordinates. Compare to the semi-recursive method, the dimension of the matrices are reduced by the velocity transformation matrix into the minimal set of relative joint coordinates. When the system of matrices is too small, the sparse technique is not practically to be implemented as several steps need to be added when solving the linear systems.

An example of the actual hydraulic circuit used in the hydraulic model is shown in Figure 4.8. The valve is modeled using a semi-empirical model in which the parameters of the valve are obtained from the manufacturer's catalogue. A short time-step is used in order to avoid numerical instability of the hydraulic simulation.



(a) Hydraulic circuit

Figure 4.8: An example of working hydraulic simulation modeling

The working hydraulic circuit consists of the hydraulic cylinder, valves, pressure relief valves and pump, as shown in Figure 4.8(a). The flowrate is controlled by the control valve spool, as shown in Figure 4.8(b), which from the simulation perspective varies between -10 and 10. The movement of valve spool will affected the pressure to be directed either to OuterBoomA or OuterBoomB. Figure 4.8(c)-(e) show the pressure rate for outer boom behavior as a reaction from the spool valve movement.

5 Conclusions

The objective of this dissertation is to introduce approaches to implement multibody methods in real-time simulation of working mobile vehicles. Multibody systems can be modeled based on rigid and flexible bodies. In this dissertation, two methods are considered in detail: an augmented Lagrangian method and a semi-recursive method based on a velocity transformation matrix. Both methods are able to produce good results within the real time periods. Additionally, the multibody model is coupled with a number of other sub-models, which are a hydraulic model, and a contact model and tire model. All of this sub-models are solved simultaneously within the required time step.

Lumped fluid theory is used to model the hydraulic circuit. This method divides the hydraulic circuit into volumes with an assumption the pressure are equally distributed. The time step use in the hydraulic modeling need to be short as the hydraulic circuit normally has high nominal frequencies. The valve is modeled using semi-empirical approach where the parameters can be obtained from the manufacture catalogues.

In contact model, the model is divided into two stages which are collision detection and collision response. Collision detection is functional to determine the time and location of the contact while collision response is to provide the forces at the contact point. Spring-damper element is added as the penalty parameters at the contact point. The friction between the tire and ground is modeled using lumped LuGre friction model. This method is employed with an assumption the friction occur at the single contact point. At the microscopic level, the contact point consist of the elastic bristles. The tangential force will deflect the bristle like a springs. The amount of the bristle deflection can be computed and represents in term of friction force.

For flexible bodies, an approach based on a floating frame of reference formulation is introduced. The deformation mode needed in the formulation can be obtained using a finite element method. Often a structural model based on the finite element method consists of large number of degrees of freedom. For this reason, model order reduction methods such as Guyan reduction, Craig-Bampton method or Krylov subspace are required to make the finite element model suitable for use with multibody system dynamics. Due to the large number of degrees of freedom that need to be processed, reduction methods can be implemented in the sparse technique using sparse solvers to determine eigenvalues and eigenvectors.

After implementing the introduced methods for mobile working vehicle models, good real-time simulation results were obtained. The semi-recursive method based on topology structure approach in particular was found to be an efficient way of modeling working mobile vehicles. In this approach, the equations of motion is based on implementation of a velocity transformation matrix with a minimal set of relative joints. The velocity transformation matrix and mass matrix can be computed in parallel, which saves computing time. For closed-loop systems, the system needs to be transformed into an open-loop system with a penalty formulation imposed at the opened joint.

Knowing the matrix structure, computational efficiency can be improved by the implementation of the sparse matrix technique. From the results, it is found this approach is suitable if the working mobile vehicle model has a large number of body, degrees of freedom and constraint equations. Therefore, it should be noted that the computational efficiency when using the sparse technique is strongly case dependent.

5.1 Future work

In flexible bodies, a practical way to determine possible deformation modes of the structure is from the finite element model. In practice, the original finite element model need to be subjected to a modal reduction procedure to make it computationally suitable for multibody applications. From this study, Krylov subspace was found to be suitable method to conduct modal reduction. However, in this study only general Arnoldi and Lanczos algorithms has been used to determine the eigenvalues and eigenvectors. There are more techniques in this method which can be explored such as Generalized Minimum Residual (GMRES) and Biconjugate Gradient Stabilized (BiCGSTAB) whereas each technique is purposely to solve specific system such as nonsymmetric linear system.

In this study for semi-recursive method, the opened closed-loop joint is imposed with the penalty method in order to avoid constraint violation. The other method which can be explored is by implementing second transformation of velocities as proposed in [4]. It is claimed to be faster and free from constraint problem.

References

[1] R. B. Gillespie and J. E. Colgate, “A survey of multibody dynamics for virtual environments,” in *International Mechanical Engineering Congress and Exhibition*, Dallas, 1997.

[2] J. Ginsberg, Engineering dynamics, Cambridge: Cambridge University Press, 2008.

[3] J. C. G. Orden, J. M. Goicolea and J. Cuadrado, Multibody dynamics: Computational methods and applications, Dordrecht: Springer Science & Business Media, 2007.

[4] J. G. de Jalon and B. Eduardo, Kinematic and dynamic simulation of multibody systems: The real-time challenge, New York: Springer Science & Business Media, 2012.

[5] A. A. Shabana, Dynamics of multibody systems, Cambridge: Cambridge University Press, 2013.

[6] P. Flores, Concepts and formulations for spatial multibody dynamics, Portugal: Springer, 2015.

[7] A. Avello, J. M. Jimenez, E. Bayo and J. G. de Jalon, “A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformations,” *Computer Methods in Applied Mechanics and Engineering,* vol. 107, no. 3, pp. 313-339, 1993.

[8] F. C. Moon, Applied dynamics: With applications to multibody and mechatronic systems, New York: John Wiley & Sons, 2008.

[9] E. Bayo and R. Ledesma, “Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics,” *Nonlinear Dynamics,* vol. 9, no. 1-2, pp. 113-130, 1996.

[10] O. A. Bauchau, Flexible multibody dynamics, vol. 176, Dordrecht: Springer Science & Business Media, 2010.

[11] A. A. Shabana, “Flexible multibody dynamics: Review of past and recent developments,” *Multibody System Dynamics,* vol. 1, no. 2, pp. 189-222, 1997.

[12] Z.-Q. Qu, Model order reduction techniques with applications in finite element analysis, London: Springer Science & Business Media, 2013.

[13] C. Nowakowski, J. Fehr, M. Fischer and P. Eberhard, “Model order reduction in elastic multibody systems using the floating frame of reference formulation,” in *7th International Conference on Mathematical Modelling*, Vienna, 2012.

[14] W. H. Press, Numerical recipes 3rd edition: The art of scientific computing, New York: Cambridge University Press, 2007.

[15] A. O. Anidu, S. A. Arekete, A. O. Adedayo and A. O. Adekoya, “Dynamic computation of Runge-Kutta’s fourth-order algorithm for first and second order ordinary differential equation using java,” *IJCSI International Journal of Computer Science Issues,* vol. 12, no. 3, pp. 211-218, 2015.

[16] P. Korkealaakso, A. Rouvinen, S. Moisio and J. Peusaari, “Development of a real-time simulation environment,” *Multibody System Dynamics,* vol. 17, no. 2-3, pp. 177-194, 2007.

[17] K. J. Astrom and C. Canudas-De-Wit, “Revisiting the LuGre friction model,” *Control Systems, IEEE,* vol. 28, no. 6, pp. 101-114, 2008.

[18] J. Watton, Fluid power system, Englewood Cliff: Prentice Hall, 1989.

[19] A. Boeing and T. Braun, “Evaluation of real-time physics simulation systems,” in *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, Perth, 2007.

[20] J. Baumgarte, “Stabilization of constraints and integrals of motion in dynamical systems,” *Computer Methods in Applied Mechanics and Engineering,* vol. 1, no. 1, pp. 1-16, 1972.

[21] E. Bayo, J. G. de Jalon and M. A. Serna, “A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems,” *Computer Methods in Applied Mechanics and Engineering,* vol. 71, no. 2, pp. 183-195, 1988.

[22] O. A. Bauchau and A. Laulusa, “Review of contemporary approaches for constraint enforcement in multibody systems,” *Journal of Computational and Nonlinear Dynamics,* vol. 3, no. 1, pp. 1-8, 2008.

[23] J. C. G. Orden, . J. M. Goicolea and J. Cuadrado, Multibody dynamics: Computational methods and applications, vol. 4, Dordrecht: Springer Science & Business Media, 2007.

[24] J. de Garcia, E. Alvarez, F. de Ribera, I. Rodriguez and F. Funes, “A fast and simple semi-recursive formulation for multi-rigid-body systems,” in *Advances in Computational Multibody Systems*, Springer, pp. 1-23, 2005.

[25] J. Jimenez, A. Avello, J. G. de Jalon and A. Avello, “An efficient implementation of the velocity transformation method for real-time dynamics with illustrative examples,” in *Computational Dynamics in Multibody Systems*, Springer, pp. 15-35, 1995.

[26] P. Korkealaakso, A. Mikkola, T. Rantalainen and A. Rouvinen, “Description of joint constraints in the floating frame of reference formulation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics,* vol. 223, no. 2, pp. 133-145, 2013.

[27] R. J. Guyan, “Reduction of stiffness and mass matrices,” *AIAA journal,* vol. 3, no. 2, p. 380, 1965.

[28] C. C. Flanigan, “Model reduction using Guyan, IRS, and dynamic methods,” in *Proceedings of 16th International Modal Analysis Conference*, Santa Barbara, 1998.

[29] O. P. Agrawal and A. A. Shabana, “Dynamic analysis of multibody systems using component modes,” *Computers & Structures,* vol. 21, no. 6, pp. 1303-1312, 1985.

[30] J. Young and W. Haile, “Primer on the Craig-Bampton method,” *Finite Element Modeling Continuous Improvement, NASA,* 2000.

[31] H. A. Van der Vorst, Iterative Krylov methods for large linear systems, vol. 13, Cambridge: Cambridge University Press, 2003.

[32] C. Lein, M. Beitelschmidt and D. Bernstein, “Improvement of Krylov-Subspace-Reduced Models by Iterative Mode-Truncation,” *IFAC-PapersOnLine,* vol. 48, no. 1, pp. 178-183, 2015.

[33] Z. Bai, “Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems,” *Applied Numerical Mathematics,* vol. 43, no. 1, pp. 9-44, 2002.

[34] D. S. Watkins, The matrix eigenvalue problem: GR and Krylov subspace methods, Philadelphia: Society for Industrial and Applied Mathematics, 2007.

[35] Y. Saad, Numerical methods for large eigenvalue problems, vol. 2, Philadelphia: Society for industrial and applied mathematics, 2011.

[36] P. J. Heres, Robust and efficient Krylov subspace methods for model order reduction, Eindhoven: Technische Universiteit Eindhoven, 2005.

[37] H. Ghozilah, Modeling and experimental evaluation of the effective bulk modulus for a mixture of hydraulic oil and air, Doctoral thesis, Saskatchewan, Canada: University of Saskatchewan, 2013.

[38] S. Ilango and V. Soundararajan, Introduction to hydraulics and pneumatics, Delhi: PHI Learning Pvt. Ltd., 2011.

[39] C. Ericson, Real-time collision detection, San Francisco: Morgan Kaufmann, 2004.

[40] S. Moisio, B. Leon, P. Korkealaakso and A. Morales, "Simulation of tactile sensors using soft contacts for robot grasping applications," in *IEEE International Conference on Robotics and Automation*, Saint Paul, 2012.

[41] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," *ACM SIGGRAPH Computer Graphics,* vol. 23, no. 3, pp. 223-232, 1989.

[42] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," *ACM SIGGRAPH Computer Graphics,* vol. 22, no. 4, pp. 289-298, 1988.

[43] P. Kraus and V. Kumar, "Compliant contact models for rigid body collisions," in *Proceedings IEEE International Conference on Robotics and Automation*, Albuerque, New Mexico, 1997.

[44] T. Piatkowski, "Dahl and LuGre dynamic friction models—The analysis of selected properties," *Mechanism and Machine Theory,* vol. 73, pp. 91-100, 2014.

[45] P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright, "Sparse matrix methods in optimization," *SIAM Journal on Scientific and Statistical Computing,* vol. 5, no. 3, pp. 562-589, 1984.

[46] I. Duff, A. Erisman and J. Reid, Direct methods for sparse matrices, numerical mathematics scientific computation, Oxford: Oxford Science Publications, 1986.

[47] H. A. Van Der Vorst, “Efficient and reliable iterative methods for linear systems,” *Journal of Computational and Applied Mathematics,* vol. 149, no. 1, pp. 251-265, 2002.

[48] R. Serban, D. Negrut, E. J. Haug and F. A. Potra, “A topology-based approach for exploiting sparsity in multibody dynamics in cartesian formulation,” *Journal of Structural Mechanics,* vol. 25, no. 3, pp. 379-396, 1997.

[49] T. A. Davis, Direct methods for sparse linear systems, vol. 2, Philadelphia: Society for Industrial and Applied Mathematics, 2006.

[50] Y. Saad, Iterative methods for sparse linear systems, Philadelphia: Society for Industrial and Applied Mathematics, 2003.

[51] P. Korkealaakso, A. J. Rouvinen, S. M. Moisio and J. Peusaari, “Development of a real-time simulation environment,” *Multibody System Dynamics,* vol. 17, no. 2-3, pp. 177-194, 2007.

[52] N. Bell and M. Garland, *Efficient sparse matrix-vector multiplication on CUDA,* Nvidia Technical Report NVR-2008-004, Nvidia Corporation, 2008.

[53] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenny, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof and D. Sorensen, “LAPACK: A portable linear algebra library for high-performance computers,” in *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, New York, 1990.

[54] T. A. Davis, UMFPACK version 4.4 user guide, Gainesville, FL: Department of Computer and Information Science and Engineering, University of Florida, 2005.

[55] R. B. Lehoucq, D. C. Sorensen and C. Yang, ARPACK users' guide: Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, Philadelphia: Society for Industrial and Applied Mathematics, 1998.

[56] I. S. Duff, M. A. Heroux and R. Pozo, “An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum,” *ACM Transactions on Mathematical Software (TOMS),* vol. 28, no. 2, pp. 239-267, 2002.

[57] X. S. Li, “An overview of SuperLU: Algorithms, implementation, and user interface,” *ACM Transactions on Mathematical Software,* vol. 31, no. 3, pp. 302-325, 2005.

Publication I

Baharudin, E., Korkealaakso, P., Rouvinen, A., and Mikkola, A.
Crane Operators Training Based on the Real-Time Multibody Simulation

Reprinted with permission from
Multibody System Dynamics, Robotics and Control
Pp. 213-229, 2013

Chapter 13
Crane Operators Training Based on the Real-Time Multibody Simulation

Mohamad Ezral Baharudin, Pasi Korkealaakso, Asko Rouvinen, and Aki Mikkola

Abstract This paper introduces a real-time multibody simulation approach. Two main sections have been described in depth and include a description of flexible bodies and modeling of a hydraulic system. In flexible bodies, the bodies are modelled using the floating frame of reference formulation. The equation of motion for the body is developed using the principle of virtual work. Penalty method is used when there are constraints in the mechanical system. The hydraulic system is modelled using lumped fluid theory. Two types of components, valves and hydraulic cylinders, are introduced for modelling. A numerical example is developed using two Craig-Bampton modes deformation modes modelled as flexible bodies.

13.1 Introduction

Products of mechanical industry, such as mobile machines and cranes, contain several different technology areas such as mechanics, actuators and control systems. These subsystems are in close interaction with each other as is shown in Fig. 13.1. In case of cranes, the actuators are often handled, in principle, as hydraulics. The hydraulic actuators are assembled on the mechanism to produce forces acting on the mechanism. The mechanism is typically a system of bodies, which converts the forces ofthe actuators into constrained motion. Electronics, together with the control algorithm that defines the way the structure behaves, can

M.E. Baharudin (✉) • A. Mikkola
Department of Mechanical Engineering, Lappeenranta University of Technology, Lappeenranta, Finland
e-mail: ezral.baharudin@lut.fi; aki.mikkola@lut.fi

P. Korkealaakso • A. Rouvinen
MeVEA Ltd, Laserkatu 6, Lappeenranta 53850, Finland
e-mail: pasi.korkealaakso@mevea.com; asko.rouvinen@mevea.com

H. Gattringer and J. Gerstmayr (eds.), *Multibody System Dynamics, Robotics and Control*,
DOI 10.1007/978-3-7091-1289-2_13, © Springer-Verlag Wien 2013

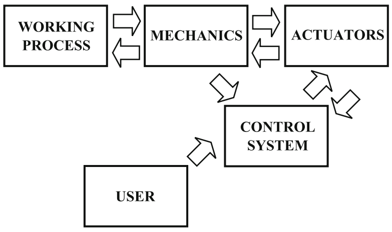


Fig. 13.1 Real-time simulation model

be used in both open and closed loop control systems and are integrated into machine systems in order to increase productivity and ergonomics.

In most cases, solution time of the traditional simulation methods used in the product development processes are not synchronized to real-time. Accordingly, a simulation of a few seconds is allowed to take several hours of real time computation. In these systems, the control signals of the simulated system must be pre-defined and, for this reason, user interaction is described more or less experimentally based on measured data. When the simulation is executed, synchronous to real-time, the operator can produce a control signal during simulation. However, real-time solution requirements often force a simplification to be made in the simulation model. In practice, the real-time model can be considered as a trade-off between efficiency and accuracy.

The objective of this paper is to introduce a general simulation approach that can be applied for the real-time simulation of hydraulically driven cranes. The introduced approach is based on the use of the floating frame of reference formulation and is coupled with the lumped fluid theory, which allows for the description of hydraulic circuits. The floating frame of reference formulation can be used together with modal reduction methods. This feature allows for the optimization of the computational efficiency, such that solution time can be synchronized with real-time. In the section of numerical examples, the introduced simulation approach is applied to create real-time simulation models for two cranes.

13.2 Description of Flexible Bodies

In this section, the description of flexible bodies is shortly explained. In this study, the flexible bodies are modeled using the floating frame of reference formulation. The formulation can be applied to bodies that experience large rigid body translations and rotations; as well as elastic deformations. The method is based on describing deformations of a flexible body with respect to a frame of reference. The frame of reference, in turn, is employed to describe large translations and rotations. The deformations of a flexible body with respect to its frame of reference can be described with a number of methods, whereas in this study, deformation is described using linear deformation modes of the body. Deformation modes can be

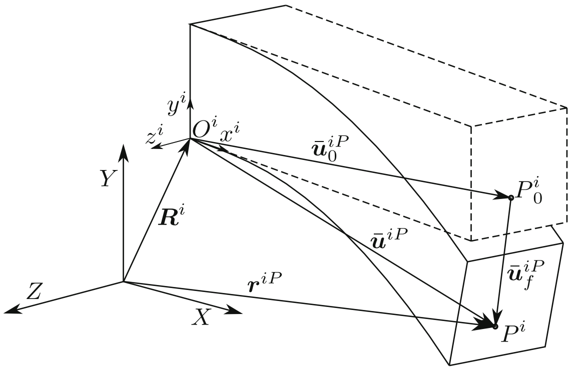

y^i
z^i
O^i
x^i
$\bar{\boldsymbol{u}}_0^{iP}$
P_0^i
Y
$\boldsymbol{R}^i$
$\bar{\boldsymbol{u}}^{iP}$
$\bar{\boldsymbol{u}}_f^{iP}$
Z
$\boldsymbol{r}^{iP}$
X
P^i

Fig. 13.2 The position of the particle P^i in global coordinate system

defined using a finite element model of the body [1, 2]. Figure 13.2 illustrates the position of particle P^i in a deformed body i.

The position of particle P^i of the flexible body i can be described in a global coordinate system using the vector $\mathbf{r}^{P^i}$ as follows:

$$\mathbf{r}^{P^i} = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^{P^i} = \mathbf{R}^i + \mathbf{A}^i (\bar{\mathbf{u}}_0^{P^i} + \bar{\mathbf{u}}_f^{P^i}) \tag{13.1}$$

where $\mathbf{R}^i$ is the position vector of the frame of reference, $\mathbf{A}^i$ is the rotation matrix of body i, $\bar{\mathbf{u}}^{P^i}$ is the position vector of particle P^i within the frame of reference, $\bar{\mathbf{u}}_0^{P^i}$ is the undeformed position vector of the particle within the frame of reference, and $\bar{\mathbf{u}}_f^{P^i}$ is the displacement of particle P^i within the frame of reference due to the deformation of body i. In this study, the rotation matrix $\mathbf{A}^i$ is expressed using Euler parameters $\theta^{E^i \mathrm{T}} = [\,\theta_0^{E^i} \quad \theta_1^{E^i} \quad \theta_2^{E^i} \quad \theta_3^{E^i}\,]^{\mathrm{T}}$ in order to avoid singular conditions which are problematic when three rotational parameters are used, such as in the case of Euler and/or Bryant angles [3]. The rotation matrix can be written using Euler parameters as follows:

$$\mathbf{A}^i = 2\begin{bmatrix} \frac{1}{2} - \left(\theta_2^{E^i}\right)^2 - \left(\theta_3^{E^i}\right)^2 & \theta_1^{E^i}\theta_2^{E^i} - \theta_0^{E^i}\theta_3^{E^i} & \theta_1^{E^i}\theta_3^{E^i} + \theta_0^{E^i}\theta_2^{E^i} \\ \theta_1^{E^i}\theta_2^{E^i} + \theta_0^{E^i}\theta_3^{E^i} & \frac{1}{2} - \left(\theta_1^{E^i}\right)^2 - \left(\theta_3^{E^i}\right)^2 & \theta_2^{E^i}\theta_3^{E^i} - \theta_0^{E^i}\theta_1^{E^i} \\ \theta_1^{E^i}\theta_3^{E^i} - \theta_0^{E^i}\theta_2^{E^i} & \theta_2^{E^i}\theta_3^{E^i} + \theta_0^{E^i}\theta_1^{E^i} & \frac{1}{2} - \left(\theta_1^{E^i}\right)^2 - \left(\theta_2^{E^i}\right)^2 \end{bmatrix} \tag{13.2}$$

The following mathematical constraint must be taken into consideration when Euler parameters are applied:

$$\left(\theta_0^{E^i}\right)^2 + \left(\theta_1^{E^i}\right)^2 + \left(\theta_2^{E^i}\right)^2 + \left(\theta_3^{E^i}\right)^2 = 1 \tag{13.3}$$

The deformation vector $\bar{\mathbf{u}}_f^{P^i}$ can be described using a linear combination of the deformation modes

$$\bar{\mathbf{u}}_f^{P^i} = \boldsymbol{\Psi}_R^{P^i}\mathbf{q}_f^i \tag{13.4}$$

Where $\boldsymbol{\Psi}_R^{P^i}$ is the modal matrix whose columns describe the translation of particle P^i within the assumed deformation modes of the flexible body i [4], and $\mathbf{q}_f^i$ is the vector of elastic coordinates. Consequently, the generalized coordinates that uniquely define the position of point P^i can be represented with vector $\mathbf{p}^i$ as follows:

$$\mathbf{p}^{i\mathrm{T}} = \begin{bmatrix} \mathbf{R}^{i\mathrm{T}} & \theta^{E^i\mathrm{T}} & \mathbf{q}_f^{i\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{13.5}$$

The velocity of particle P^i can be obtained by differentiating the position description (13.1) with respect to time

$$\dot{\mathbf{r}}^{P^i} = \dot{\mathbf{R}}^i - \mathbf{A}^i\left(\tilde{\bar{\mathbf{u}}}_0^{P^i} + \tilde{\boldsymbol{\Psi}}_R^{P^i}\mathbf{q}_f^i\right)\bar{\omega}^i + \mathbf{A}^i\boldsymbol{\Psi}_R^{P^i}\dot{\mathbf{q}}_f^i \tag{13.6}$$

Where $\bar{\omega}^i$ is the vector of local angular velocities. In (13.6), the generalized velocity vector can be defined

$$\dot{\mathbf{q}}^{i\mathrm{T}} = \begin{bmatrix} \dot{\mathbf{R}}^{i\mathrm{T}} & \bar{\omega}^{i\mathrm{T}} & \dot{\mathbf{q}}_f^{i\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{13.7}$$

By differentiating (13.6) with respect to time, the following formulation for the acceleration of particle P^i can be obtained:

$$\ddot{\mathbf{r}}^{P^i} = \ddot{\mathbf{R}}^i + \mathbf{A}^i\tilde{\bar{\omega}}^i\tilde{\bar{\omega}}^i\bar{\mathbf{u}}^{P^i} + \mathbf{A}^i\dot{\tilde{\bar{\omega}}}^i\bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i\tilde{\bar{\omega}}^i\dot{\bar{\mathbf{u}}}^{P^i} + \mathbf{A}^i\ddot{\bar{\mathbf{u}}}^{P^i} \tag{13.8}$$

where $\tilde{\bar{\omega}}^i$ is a skew-symmetric representation of the angular velocity of the body in the frame of reference, $\ddot{\mathbf{R}}^i$ is the vector that defines the translational acceleration of the frame of reference, $\mathbf{A}^i\tilde{\bar{\omega}}^i\tilde{\bar{\omega}}^i\bar{\mathbf{u}}^{P^i}$ is the normal component of acceleration, $\mathbf{A}^i\dot{\tilde{\bar{\omega}}}^i\bar{\mathbf{u}}^{P^i}$ is the tangential component of acceleration, $2\mathbf{A}^i\tilde{\bar{\omega}}^i\dot{\bar{\mathbf{u}}}^{P^i}$ is the Coriolis component of acceleration and $\mathbf{A}^i\ddot{\bar{\mathbf{u}}}^{P^i}$ is the acceleration of particle P^i due to the deformation of body i.

When deformation modes are used with the floating frame of reference, rotations due to body deformation are usually ignored. However, in order to compose all of the basic constraints, rotation due to body deformation must be accounted for. The vector $\bar{\mathbf{v}}_f^i$ due to deformation at the location of particle P^i within the frame of reference can be expressed

$$\bar{\mathbf{v}}_f^i = \mathbf{A}_f^{P^i}\bar{\mathbf{v}}^i \tag{13.9}$$

Where $\bar{\mathbf{v}}^i$ is defined in the undeformed state at the location of particle P^i, and $\mathbf{A}_f^{P^i}$ is a rotation matrix that describes the orientation due to deformation at the location of particle P^i with respect to the reference frame. Note that all components in (13.9) are expressed in the reference frame. The rotation matrix $\mathbf{A}_f^{P^i}$ can be expressed as follows:

$$\mathbf{A}_f^{P^i} = \mathbf{I} + \tilde{\varepsilon}^{P^i} \tag{13.10}$$

In (13.10), $\mathbf{I}$ is a (3 × 3) identity matrix and $\tilde{\varepsilon}^{P^i}$ is a skew symmetric form of the rotation change caused by deformation. Rotation changes due to deformation can be represented as the following:

$$\varepsilon^{P^i} = \mathbf{\Psi}_0^{P^i} \mathbf{q}_f^i \tag{13.11}$$

Where $\mathbf{\Psi}_0^{P^i}$ is the modal transformation matrix whose columns describe rotation coordinates of point P^i within the assumed deformation modes of the flexible body i [4].

13.2.1 Equations of Motion

The equations of motion can be developed using the principle of virtual work. When the floating frame of reference formulation is employed the virtual work done by inertial forces can be written as follows:

$$\delta W^{i^i} = \int_{V^i} \rho^i \delta\, \mathbf{r}^{P^i \mathrm{T}}\, \ddot{\mathbf{r}}^{P^i}\, dV^i \tag{13.12}$$

where $\delta \mathbf{r}^{P^i}$ is the virtual displacement of the position vector of a particle, $\ddot{\mathbf{r}}^{P^i}$ is the acceleration vector of the particle defined in (13.8), ρ^i is density of body i, and V^i is volume of body i. The virtual displacement of the position vector can be expressed in terms of virtual displacement of generalized coordinates as follows:

$$\delta\, \mathbf{r}^{P^i \mathrm{T}} = \begin{bmatrix} \delta \mathbf{R}^{i\mathrm{T}} & \delta\theta^{i\mathrm{T}} & \delta \mathbf{q}_f^{i\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\tilde{\bar{\mathbf{u}}}^{P^i \mathrm{T}} \;\; \mathbf{A}^{i\mathrm{T}} \\ \mathbf{\Psi}_R^{P^i \mathrm{T}} \;\; \mathbf{A}^{i\mathrm{T}} \end{bmatrix} \tag{13.13}$$

By substituting the virtual displacement of position vector (13.13) into the equation of virtual work of the inertial forces (13.12), and by separating the terms related to acceleration from the terms related quadratically to velocities, the following equation for the virtual work of inertial forces can be obtained:

$$\delta W^{i^i} = \delta \mathbf{q}^i \left[\mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{v^i} \right] \tag{13.14}$$

Where $\mathbf{M}^i$ is the mass matrix and $\mathbf{Q}^{v^i}$ is the quadratic velocity vector. The mass matrix can be expressed as follows:

$$\mathbf{M}^i = \int\limits_{V^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A}^i \tilde{\bar{\mathbf{u}}}^{P^i} & \mathbf{A}^i \Psi_R^{P^i} \\ & \tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\mathbf{u}}}^{P^i} & -\tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \Psi_R^{P^i} \\ sym & & \Psi_R^{P^i\mathrm{T}} \Psi_R^{P^i} \end{bmatrix} dV^i \tag{13.15}$$

And, correspondingly, the quadratic velocity vector takes the form

$$\mathbf{Q}^{v^i} = \int\limits_{V^i} \rho^i \begin{bmatrix} \mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \\ -\tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} - 2\tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \\ \Psi_R^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\Psi_R^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \Psi_R^{P^i} \dot{\mathbf{q}}_f^i \end{bmatrix} dV^i \tag{13.16}$$

The virtual work of the externally applied forces can be written as:

$$\delta W^{e^i} = \int\limits_{V^i} \delta \mathbf{r}^{P^i\mathrm{T}} \mathbf{F}^{P^i} dV^i = \delta \mathbf{q}^{i\mathrm{T}} \mathbf{Q}^{e^i} \tag{13.17}$$

Where $\mathbf{F}^{P^i}$ is external force per unit mass and $\mathbf{Q}^{e^i}$ is the vector of generalized forces which can be expressed as follows:

$$\mathbf{Q}^{e^i} = \begin{bmatrix} \sum\limits_{j=1}^{n_F} \mathbf{F}_j^i \\ \sum\limits_{j=1}^{n_F} \tilde{\bar{\mathbf{u}}}_j^i \mathbf{A}^{i\mathrm{T}} \mathbf{F}_j^i \\ \sum\limits_{j=1}^{n_F} \Psi_j^{i\mathrm{T}} \mathbf{A}^{i\mathrm{T}} \mathbf{F}_j^i \end{bmatrix} \tag{13.18}$$

where $\mathbf{F}_j^i$ is the j-th force component acting on body i, $\tilde{\bar{\mathbf{u}}}_j^i$ is a skew symmetric matrix of the location vector of the j-th force components, and Ψ_j^i includes the terms of the modal matrix associated with the node to which the j-th force component applies.

The elastic forces can be defined using the modal stiffness matrix $\mathbf{K}^i$ and modal coordinates. The modal stiffness matrix is associated with the modal coordinates and the matrix can be obtained from the conventional finite element approach using the component mode synthesis technique [4]. The virtual work of elastic forces can be written as follows:

$$\delta W^{s^i} = \delta \mathbf{q}_f^{i\mathrm{T}} \mathbf{K}^i \mathbf{q}_f^i \tag{13.19}$$

Accordingly, the vector of elastic forces can be represented as follows:

$$\mathbf{Q}^{f^i} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix} \tag{13.20}$$

Using (13.14), (13.17) and (13.19), the equation of virtual work, including inertial, external and internal force components, can be written as follows:

$$\delta \mathbf{q}^i \left[\mathbf{M}^i \ddot{\mathbf{q}}^i + \mathbf{Q}^{v^i} + \mathbf{Q}^{f^i} - \mathbf{Q}^{e^i} \right] = 0 \tag{13.21}$$

The terms inside the brackets can be used to form unconstrained Newton-Euler equations as follows:

$$\begin{aligned}
&\begin{bmatrix} \int\limits_{V^i} \rho^i \mathbf{I} dV^i & -\int\limits_{V^i} \rho^i \mathbf{A}^i \tilde{\bar{\mathbf{u}}}^{P^i} dV^i & \int\limits_{V^i} \rho^i \mathbf{A}^i \boldsymbol{\Psi}_R^{P^i} dV^i \\ & \int\limits_{V^i} \rho^i \tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\mathbf{u}}}^{P^i} dV^i & -\int\limits_{V^i} \rho^i \tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \boldsymbol{\Psi}_R^{P^i} dV^i \\ sym & & \int\limits_{V^i} \rho^i \boldsymbol{\Psi}_R^{P^i\mathrm{T}} \boldsymbol{\Psi}_R^{P^i} dV^i \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}}^i \\ \dot{\bar{\omega}}^i \\ \ddot{\mathbf{q}}_f^i \end{bmatrix} \\
&= \begin{bmatrix} \int\limits_{V^i} \mathbf{F}^{P^i} dV^i \\ \int\limits_{V^i} \tilde{\bar{\mathbf{u}}}^{P^i} \mathbf{A}^{i\mathrm{T}} \mathbf{F}^{P^i} dV^i \\ \int\limits_{V^i} \boldsymbol{\Psi}_R^{P^i\mathrm{T}} \mathbf{A}^{i\mathrm{T}} \mathbf{F}^{P^i} dV^i \end{bmatrix} \\
&- \begin{bmatrix} \int\limits_{V^i} \rho^i \left(\mathbf{A}^i \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\mathbf{A}^i \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \\ \int\limits_{V^i} \rho^i \left(-\tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} - 2\tilde{\bar{\mathbf{u}}}^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \\ \int\limits_{V^i} \rho^i \left(\boldsymbol{\Psi}_R^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \tilde{\bar{\omega}}^i \bar{\mathbf{u}}^{P^i} + 2\boldsymbol{\Psi}_R^{P^i\mathrm{T}} \tilde{\bar{\omega}}^i \boldsymbol{\Psi}_R^{P^i} \dot{\mathbf{q}}_f^i \right) dV^i \end{bmatrix} \\
&- \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}^i \mathbf{q}_f^i \end{bmatrix}
\end{aligned} \tag{13.22}$$

Equations of motion in this form are referred to as Generalized Newton-Euler equations in [4], where Newton-Euler equations of rigid bodies are extended to flexible bodies.

13.2.2 Integration of the Equations of Motion

Due to the use of Generalized Newton-Euler equations as a description of dynamics, the equations of motion are expressed using the angular velocity and angular acceleration vectors. Equation (13.22) can be solved to obtain angular accelerations in the body frame, which can be integrated with angular velocities. However, the problem arises when the coordinates describing the orientation of the body have to be solved. This is due to the fact that angular velocities cannot be directly integrated with the parameters which uniquely describe the orientation of the body. For this reason, a new set of variables **p** is defined, containing the orientation coordinates of the body reference frame. In order to integrate the position level coordinates, a relationship between the first time derivative of Euler parameters and the vector of angular velocities defined in the body reference frame can be made through the following linear expression:

$$\dot{\theta}^{E^i} = \frac{1}{2}\bar{\mathbf{G}}^{i\,\mathrm{T}}\bar{\omega}^i \tag{13.23}$$

where the velocity transformation matrix $\bar{\mathbf{G}}^i$ can be written as follows:

$$\bar{\mathbf{G}}^i = \begin{bmatrix} -\theta_1^{E^i} & \theta_0^{E^i} & \theta_3^{E^i} & -\theta_2^{E^i} \\ -\theta_2^{E^i} & -\theta_3^{E^i} & \theta_0^{E^i} & \theta_1^{E^i} \\ -\theta_3^{E^i} & \theta_2^{E^i} & -\theta_1^{E^i} & \theta_0^{E^i} \end{bmatrix} \tag{13.24}$$

The time derivatives of the body variables to be integrated can be stated using vector $\dot{\mathbf{p}}$ as follows:

$$\dot{\mathbf{p}}^{i\mathrm{T}} = \left[\dot{\mathbf{R}}^{i\mathrm{T}}\dot{\theta}^{E^i\mathrm{T}}\dot{\mathbf{q}}_f^{i\,\mathrm{T}}\right]^{\mathrm{T}} \tag{13.25}$$

which can be integrated to obtain position level generalized coordinates **p**.

13.2.3 Description of Constrained Mechanical Systems

In this section, the penalty method used in this study is briefly described. Mechanical joints that restrict motion possibilities of interconnected bodies can be described with constraint equations [5]. To satisfy a set of m constraint equations related to generalized coordinates, the following equation must be fulfilled:

$$\mathbf{C}(\mathbf{q}, t) = 0 \tag{13.26}$$

where **C** is a vector of constraints of the system and t is time. In the penalty method, the constraints are combined to the equations of motion by employing penalty terms. This procedure leads to a set of n differential equations as follows:

$$\left(\mathbf{M} + \mathbf{C}_q^{\mathrm{T}} \alpha \mathbf{C}_q\right) \ddot{\mathbf{q}} = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f - \mathbf{C}_q^{\mathrm{T}} \alpha \left(\mathbf{Q}^c + 2\Omega\mu\dot{\mathbf{C}} + \Omega^2\mathbf{C}\right) \tag{13.27}$$

where $\mathbf{C}_q$ is the Jacobian matrix of the constraint equations and α, Ω and μ are $m \times m$ diagonal matrices and contain penalty terms, natural frequencies and damping ratios for constraints, respectively. If the penalty terms are equivalent to each constraint, the matrices are identity matrices multiplied with a constant penalty factor [6].

A drawback associated with the penalty method is that large numerical values for penalty factors must be used, which may lead to numerical ill-conditioning and round-off errors. However, the method can be improved by adding penalty terms or correction terms, which are zero when constraint equations are fulfilled. Using this approach, equations of motion can be written as follows:

$$\left(\mathbf{M} + \mathbf{C}_q^{\mathrm{T}} \alpha \mathbf{C}_q\right) \ddot{\mathbf{q}} = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f - \mathbf{C}_q^{\mathrm{T}} \alpha \left(\mathbf{Q}^c + 2\Omega\mu\dot{\mathbf{C}} + \Omega^2\mathbf{C}\right) + \mathbf{C}_q^{\mathrm{T}} \lambda^* \tag{13.28}$$

where λ^* is the vector of penalty forces. Since the exact values of λ^* are not known in advance, an iterative procedure should be used as follows:

$$\lambda_{i+1}^* = \lambda_i^* - \alpha\left(\mathbf{C}_q \ddot{\mathbf{q}}_i + \mathbf{Q}^c + 2\Omega\mu\dot{\mathbf{C}} + \Omega^2\mathbf{C}\right) \tag{13.29}$$

where $\lambda^* = 0$ is used for the first iteration. Using this equation, the forces caused by errors in constraint equations at iteration $i + 1$ can be defined and compensated. In this case, the penalty terms do not need to have large numerical values. An iterative procedure can be applied directly to (13.28), which leads to the following expression:

$$\left(\mathbf{M} + \mathbf{C}_q^{\mathrm{T}} \alpha \mathbf{C}_q\right) \ddot{\mathbf{q}}_{i+1} = \mathbf{M}\ddot{\mathbf{q}}_i - \mathbf{C}_q^{\mathrm{T}} \alpha \left(\mathbf{Q}^c + 2\Omega\mu\dot{\mathbf{C}} + \Omega^2\mathbf{C}\right) \tag{13.30}$$

In the case of the first iteration, $\mathbf{M}\ddot{\mathbf{q}}_0 = \mathbf{Q}^e - \mathbf{Q}^v - \mathbf{Q}^f$, the leading matrix of (13.29) is a symmetric and positive definite, which makes the solution of the equation efficient. This formulation behaves satisfactorily also in singular configurations of a mechanical system.

13.3 Modelling of Actuators

In this study, crane systems are assumed to be driven with hydraulic actuators. Hydraulic actuators can be modelled using the lumped fluid theory, in which the hydraulic circuit is divided into volumes where pressures are assumed to be equally distributed. In this approach, the pressure waves in pipes and hoses are assumed to be insignificant [7]. The hydraulic pressure in each hydraulic volume i can be described as

$$\dot{p}_l = \frac{B_{ei}}{V_i} \sum_{j=1}^{n_c} Q_{ij} \tag{13.31}$$

where B_{ei} is the effective bulk modulus of volume i, (13.31), Q_{ij} is the outgoing or incoming flow rate of volume i and n_c is the total number of flows related to volume i. The effective bulk modulus defines the flexibility of the hydraulics and it can be calculated as follows:

$$B_{ei} = \frac{1}{\frac{1}{B_{oil}} + \sum_{j=1}^{n_c} \frac{V_j}{V_i B_j} + \sum_{k=1}^{n_h} Q_{ij} \frac{V_k}{V_i B_k}} \tag{13.32}$$

In (13.32), n_h is the total number of pipes and hoses related to volume i. The bulk modulus of oil, B_{oil}, accounts for the amount of non-dissolving air in oil and it is a function of pressure. The maximum value is typically $B_{oilmax} = 1.6e9Pa$. It is important to note that the bulk modulus B_j of component j is also dependent on the component type.

13.3.1 Modelling of Valves

For modelling purposes, a valve is assumed to consist of several adjustable restrictor valves which can each be modelled separately [8]. With small pressure differences (pressure difference $<$ 1 bar), the flow over the restrictor is thought to be laminar, whereas with larger differences, it is thought to be turbulent. When using the semi-empiric modelling method, the flow over the restrictor can be written as follows:

$$Q = C_v U \sqrt{dp} \tag{13.33}$$

where C_v is the semi-empiric flow rate constant and defines the size of the valve and U is a variable that defines the spool or poppet position. For a number of valve types, the variable U can be defined using a first order differential equationas follows:

$$\dot{U} = \frac{U_{ref} - U}{\tau} \tag{13.34}$$

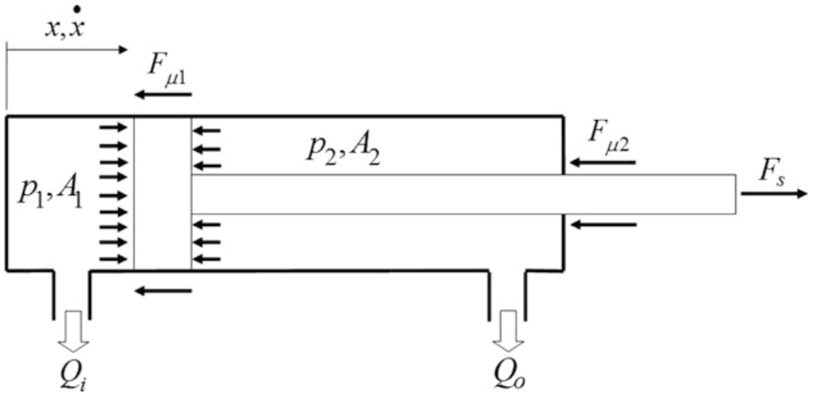


Fig. 13.3 Hydraulic cylinder transform hydraulic pressure into a mechanical force

where U_{ref} is the spool reference position and t time constant describing the dynamics of the valve spool.

13.3.2 Modelling of Hydraulic Cylinders

A hydraulic cylinder can be modeled by simply using the dimensions of the cylinder and the pressure obtained from (13.31). The motion of the hydraulic cylinder produces a flow rate to the hydraulic volume I as follows

$$\begin{aligned} Q_{jA} &= -\dot{x}A_A \\ Q_{jB} &= \dot{x}A_B \end{aligned} \tag{13.35}$$

where A_A is the area of cylinder piston side and A_B is the area on cylinder piston rod side, $\dot{x}$ is the velocity of the stroke of the cylinder, as depicted in Fig. 13.3.

The force produced by the hydraulic cylinder can be written as

$$F_s = p_1A_1 - p_2A_2 - F_\mu \tag{13.36}$$

where F_μ is the total friction force of the cylinder and p_1 and p_2 are pressures acting in the cylinder chambers. The friction force is a function of pressures, cylinder efficiency, η and velocity. The friction force can be described in a simple case as follows:

$$F_\mu = (p_1A_1 - p_2A_2)(1-\eta)f(\dot{x}) \tag{13.37}$$

The velocity dependent co-efficient, $f(\dot{x})$, can be described using a spline-curve as shown in Fig. 13.4.

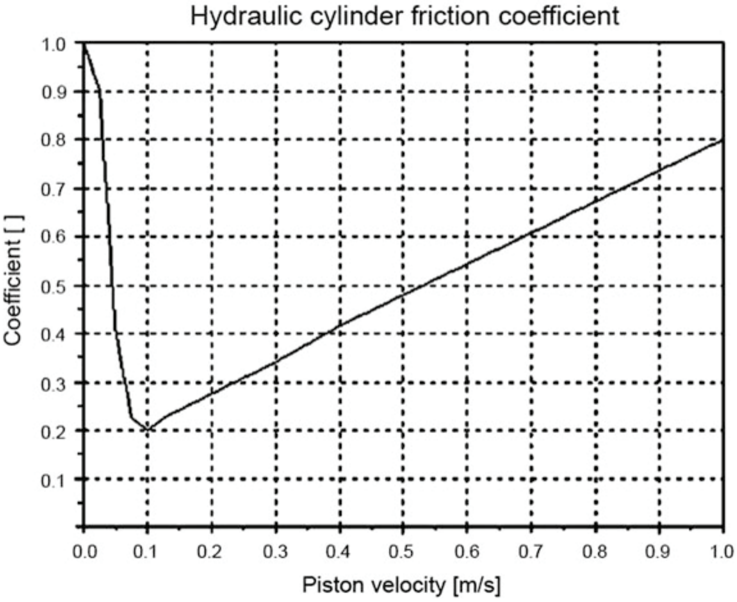


Fig. 13.4 The velocity dependent friction co-efficient

13.3.3 Numerical Example

The numerical example is based on a Liebherr LTM 1030 mobile crane and is shown in Fig. 13.5. The simulation model of the crane consists of eight bodies, of which four are modelled as flexible bodies using two Craig-Bampton deformation modes for each body. The model includes several force components related to wind loads of hooks and booms, hydraulics, the engine and a rope system using a particle based approach is modelled as well.

Since the model under investigation is used in training simulators, the hydraulic model is a simplified version of the actual hydraulics shown in Fig. 13.6.

Hydraulic circuits consist of hydraulic cylinders, pressure compensated proportional directional valves, pressure relief valves and pumps – in case of lift circuits, two counter balance valves are used. Figures 13.7 and 13.8 show examples of a simple work cycle, in Fig. 13.7 the valve control signal and spool opening are presented. Figure 13.8 presents the flow rates through the lift circuit valve and the pressure rates of cylinder chambers are shown in Fig. 13.9.

The simulation of the system was modelled using MeVEA Real-Time Simulation environment. The environment offers the possibility of off-line simulation for more detailed models or, alternatively, real-time simulation and visualization for simplified models [9]. The environment is compatible with MeVEA Full Mission Solution, which offers a motion platform and visualization environment combined with user interface – including an operators seat, joysticks, and pedal etc. case specific instrumentation [10].

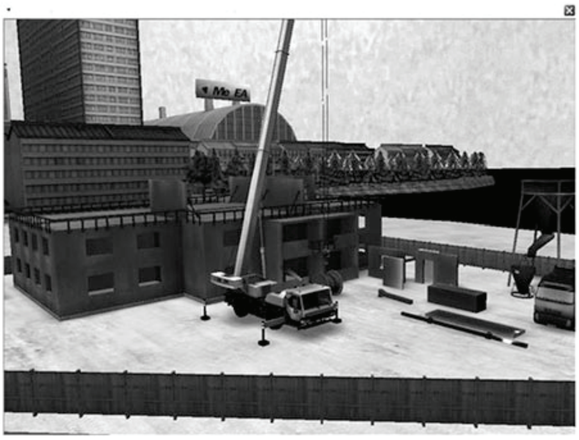


Fig. 13.5 Mobile crane in virtual construction site

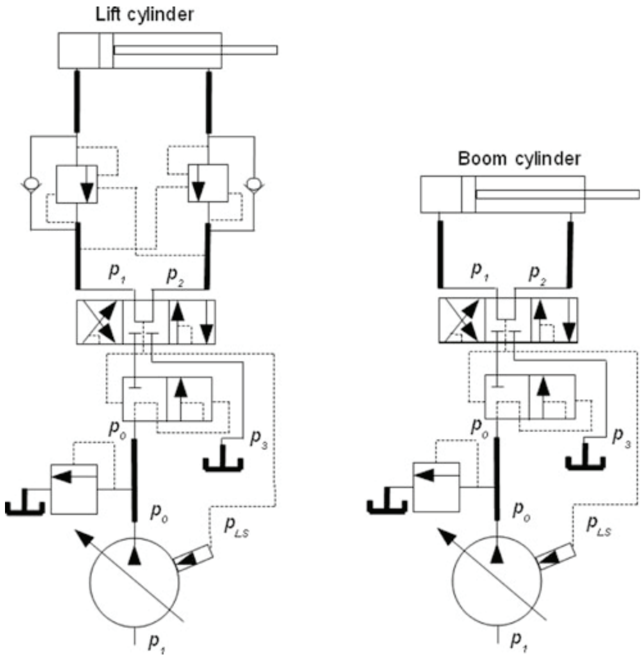


Fig. 13.6 Simplified hydraulic system of a mobile crane

13.4 Simulation Environment

In order to simulate the actual environment of the mobile crane, the dynamic model must be presented with visualization and physical feedback. This implementation required a number of mechanical actuators, interfaces, software and a high performance computer as shown in Fig. 13.10.

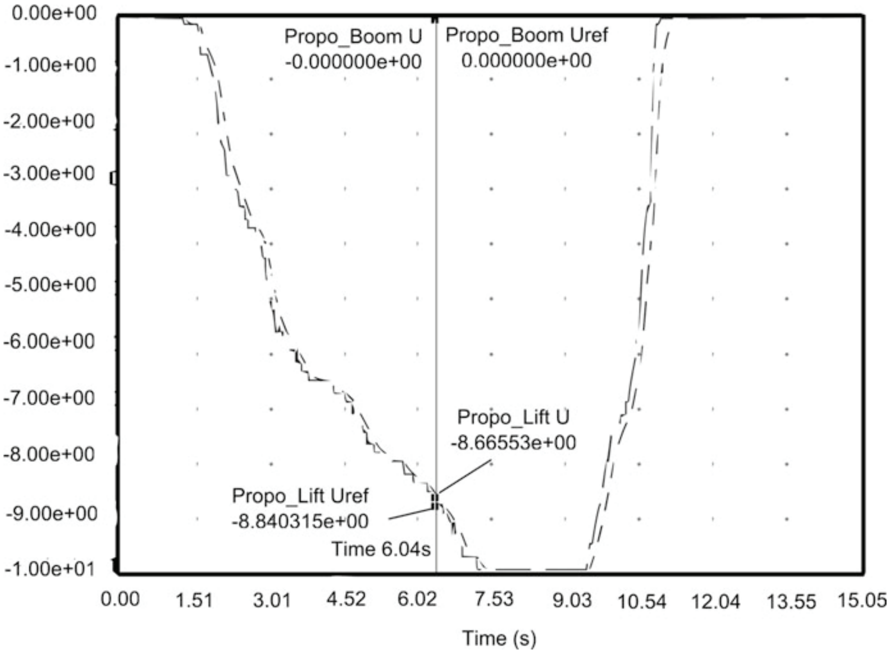


Fig. 13.7 Lift circuit valve control signal and reference control signal

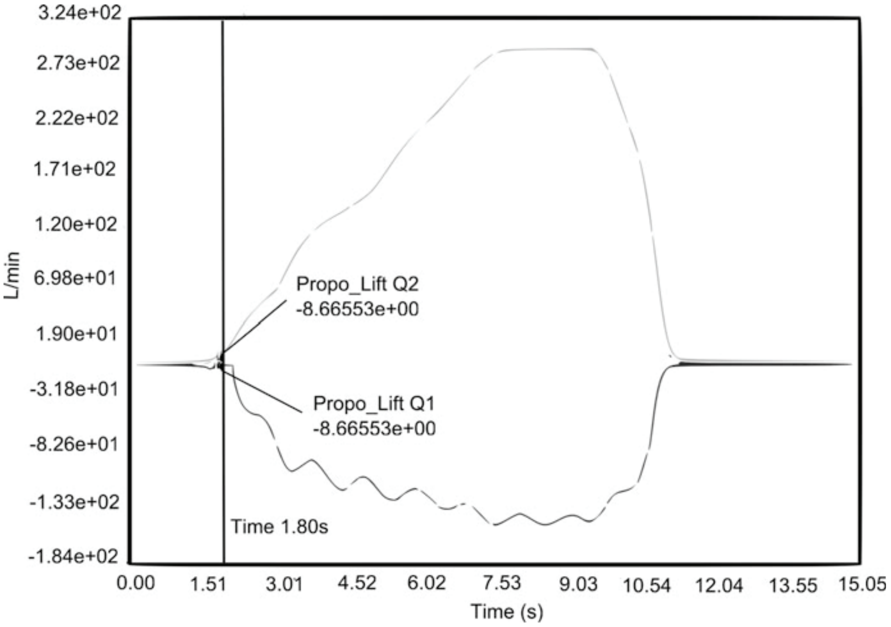


Fig. 13.8 Flow rates through the lift valve during work cycle

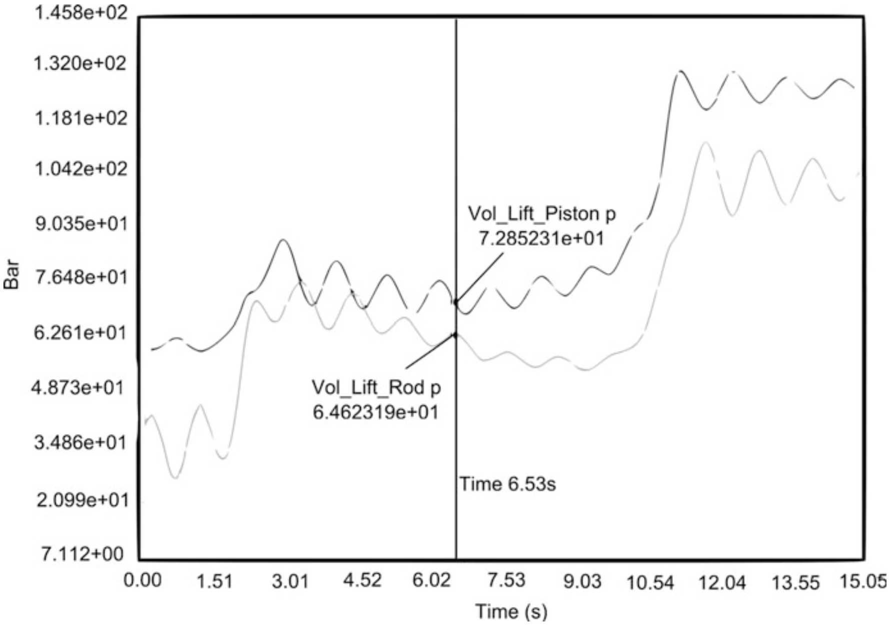


Fig. 13.9 Lift circuit cylinder pressure rates

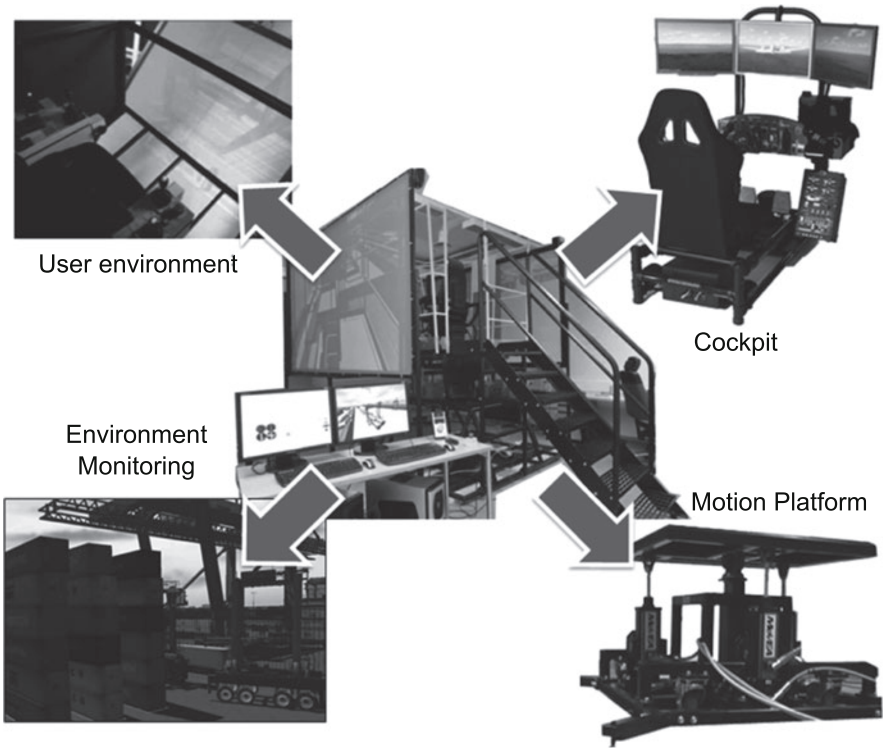


Fig. 13.10 Real-time simulation environment

The high performance computer is required to solve equations of motion and hydraulic circuits in real-time. The simulation results can be sent out to the actuators of a motion platform via I/O interfaces and a computer. These are the most critical aspects, where the response time should be minimized in order to sustain the feel and sensation of real-time feedback. The accuracy of feedback and parameters involved in the numerical analysis should also be taken into account.

The visualization of the working environment was carried out using 3D development software. All the machine components were designed with the actual dimensions, and to get a realistic result, the actual environment images were also added. All the components were treated separately in such that they have their own dedicated local coordinate system. These local coordinate systems were synchronised with the dynamic model system in a computer and were connected to another computer, which acted as a main controller for receiving and controlling all of the input and output data. The main controller reacted based on the calculation obtained from the dynamic model. It controls the position and velocity of the actuators, as well as visualization aspects.

The simulator motion platform used in this project has six DOFs. This platform has translational and rotational movements. Additional actuator may require a high performance computer in order to ensure that the modelling process is sufficient enough to generate fast data. Accurate inverse kinematic models of the platform also need to be developed to get correct acceleration rates.

13.5 Conclusion

In the real-time simulation, a machine needs to be considered as a coupled system that consists of mechanical components and actuators. This study introduced a general simulation approach that can be applied for the real-time simulation of hydraulically driven cranes. The introduced approach was based on the use of the floating frame of reference formulation and was coupled with the lumped fluid theory for the modeling of hydraulic circuits. The floating frame of reference formulation can be used together with modal reduction methods. The introduced simulation approach was applied to create real-time simulation models of a mobile crane. The simulation model of the crane consists of eight bodies, of which four are modelled as flexible bodies using two Craig-Bampton deformation modes for each body. The model includes several force components related to the wind load of hooks and booms, hydraulics, the engine and modelling of rope system using a particle based approach. The simulation model of the crane was embedded into real-time simulation environment that consists of visualization, motion platform and an I/O interface. The real-time simulation environment allows for merging a user to the simulation model in a realistic manner.

References

1. Haug EJ, Yoo WS (1986) Dynamics of flexible mechanical systems using vibration and static correction modes. J Mech Transm Automat Design 108:315–322
2. Haug EJ, Yoo WS (1986) Dynamics of articulated structures. Part I. Theory. J Struct Mech 14(1):105–126
3. Chung IS, Nikravesh PE (1982) Application of Euler parameters to the dynamic analysis of three-dimensional constrained mechanical systems. J Mech Des 104(4):785–791
4. Shabana AA (2005) Dynamics of multibody systems, 3rd edn. Cambridge University Press, Cambridge
5. Korkealaakso P, Mikkola A, Rantalainen T (2009) Description of joint constraints in the floating frame of reference formulation. Proc Inst Mech Eng Part K – J Multi-Body Dyn 223 (2):133–145
6. Bayo E, Garcia de Jalon J, Serna MA (1988) A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Comput Methods App Mech Eng 71 (2):183–195
7. Watton J (1989) Fluid fower systems. Prentice Hall International (UK) Ltd, New York, p 490
8. Handroos HM, Vilenius MJ (1991) Flexible semi-empirical models for hydraulic flow control valves. J Mech Des 113(3):232–238
9. Rouvinen A, Lehtinen T, Korkealaakso P (2005) Container gantry crane simulator for operator training. Proc Inst Mech Eng Part K – J Multi-Body Dyn 219(4):325–336
10. Korkealaakso PM, Rouvinen AJ, Moisio SM (2007) Development of a real-time simulation environment. Multibody Syst Dyn 17(2–3):177–194

Publication II

Baharudin, E., Rouvinen, A., Korkealaakso, P., and Mikkola, A.
Tree Harvester Truck Simulator based on Real-Time Multibody Simulation

Reprinted with permission from
ECCOMAS Thematic Conference on Multibody Dynamics 2013
Pp. 359-374, 2013

ECCOMAS Multibody Dynamics 2013
1-4 July, 2013, University of Zagreb, Croatia

Tree Harvester Truck Simulator based on the Real-time Multibody Simulation

Mohamad Ezral Baharudin [1], Asko Rouvinen[2], Pasi Korkealaakso[2], Aki Mikkola[1]

[1] Dept. of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland, email: ezral.baharudin@lut.fi, aki.mikkola@lut.fi
[2] MeVEA Ltd., Laserkatu 6, 53850 Lappeenranta, Finland, email: asko.rouvinen@mevea.com, pasi.korkealaakso@mevea.com

Abstract

In this study, a tree harvester truck real-time simulator has been developed to train the operator on using the truck and the cutter and managing the tree. A recursive formulation is used to express the equations of motion of the constrained mechanical system. Hydraulic actuators are modelled using the lumped fluid theory, in which the hydraulic circuit is divided into volumes where pressures are assumed to be equally distributed. In this approach, the pressure waves in pipes and hoses are assumed to be insignificant. For modelling purposes, a valve is assumed to consist of a number of adjustable restrictors which can each be modelled separately. Used contact model is divided into two parts, namely collision detection and response. Collision detection is used to identify whether moving bodies may potentially come in contact, and if so, when and where. Collision response is used to prevent penetration when contact occurs. In this study, a penalty method is used to determine collisions between objects.

Keywords: *Simulator, real-time, recursive, lumped fluid theory, contact modeling*

1. Introduction

To fulfill customer requirements, a product needs to be analyzed by coupling all the relevant engineering areas together with the number of changes made during the design phase. At the same time, competition on the market is forcing companies to shorten their product development cycles and reduce their product development costs. This, in turn, is forcing companies to minimize the number of traditional physical prototypes being used in the design phase. The drawbacks of physical prototypes are the costs associated with manufacturing unique components, the manual assembly of each prototype, the installation of measurement instruments and finally the measurements that have to be made under realistic working conditions. Therefore, the best option is to build and analyze the machine virtually. To this end, the simulation system should be able to describe the working environment and machine functions accurately to provide real-life kind of feeling to the operator. A number of tree harvester truck simulators have been developed in past to train the operator on using the truck and the cutter and managing the tree. Several well-known companies have developed such simulators, for example John Deere Forestry Machine Simulator, Valmet Komatsu Forest simulator and Creanex Training Simulator.

In the past, a simulator required a large space as well as expensive and high skills teams to manage its setup and maintenance [1]. However, with the increase of computer technology and the knowledge of multibody system, the simulator is becoming simpler and more users friendly. In practice, common real-time simulation for a simulator should consist of description of several technologies such as mechanics, actuators and a control system. E. Papadopoulos (1997) has implemented conventional linear graph method which to develop the models of the mechanical structure and actuators of harvester machine manipulator [2]. Linear graph method was developed to study the systems topology and modeling the physical systems whereas each single component is identified together with it constitutive equations. Linear graph for each subsystem could provide detail variables involve in the system and can be connected with other subsystem if those subsystem sharing the same variable. The advantage of this approach in multibody system is it is easy to switch from a relative coordinate to an absolute coordinate formulation. However, the particular efficiency of dynamic model for 3D model produced by this approach is slightly not significant compare to recursive multibody formulation [3,4]

The objective of this paper is to introduce an accurate simulation approach which closes to the real environment and feedback. This approach will be implemented to the tree harvester truck simulator and the modeling of mechanics (kinematics and dynamics), actuators, contact and control will be discussed in depth. The dynamic simulation of a mechanical system will be modeled using a semi recursive formulation with parallel computing approach. This method will be used to recursively form an equation of motion for an open loop (chain topology) rigid body system. The hydraulic actuators will be modeled using the lumped fluid theory in which the hydraulic circuit is divided into volumes where pressures are assumed to be equally distributed. The penalty method is applied to model collision and contact. In the section on the numerical example, the introduced simulation approach is applied to create real-time simulation models for a tree harvester truck.

2. Multibody Systems

Multibody systems consist of a collection of bodies that are constrained to move relative to one another by kinematic connections. The system can have a tree or chain structure (open loop) or a structure with closed links of bodies (closed loop). In the chain structure system, the path from an arbitrary body in the system to another arbitrary body in the system is unique. To make it easy to understand, a tree harvester truck can be assumed and simplified into a chain structure concept and in general it can be divided into several bodies as shown in Figure 1.

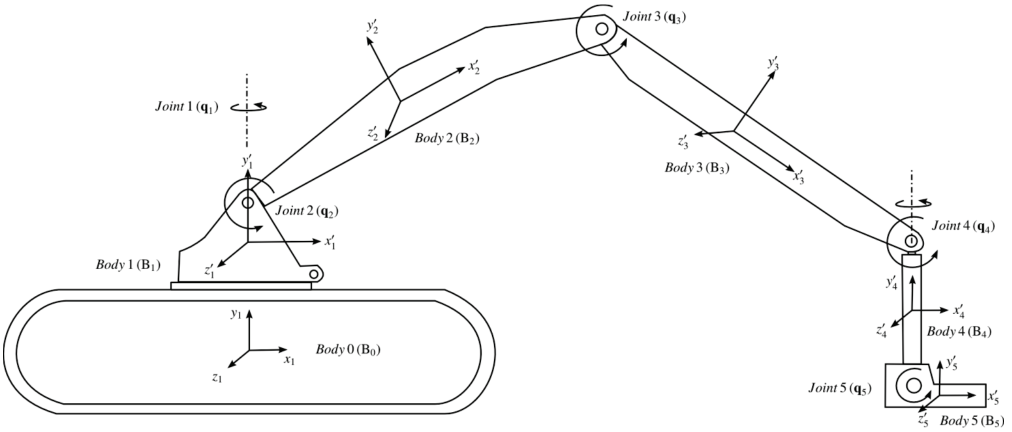


Figure 1. Tree harvester truck illustrated as rigid bodies

The tree harvester truck mechanism consists of 5 bodies, not including the vehicle body. Bodies 1, 2, 3, 4 and 5 are connected to the previous bodies by revolute joints. Vehicle body, B$_0$ is designated as the base body. Body B$_1$ is attached to body B$_0$ by a revolute joint. Joint 4 consists of two revolute joints which are vertical and horizontal. Therefore, this mechanical system has 6 degrees of freedom. Due to the movements in the horizontal and vertical, the motion of the mechanism is three dimensional.

2.1 Recursive Formalism

In this study, a recursive formulation is derived from the equation of motion for a constrained mechanical system. The kinematic properties such as position, virtual displacement, velocity and acceleration are developed based on the relative coordinates between contiguous bodies which are connected by a joint [5]. Hooker [6] proposed this formalism to the dynamic analysis of satellites and found the computational costs to increase linearly with the number of bodies. This algorithm has been used and extended by several researchers, and recently, this formalism has been generalized to improve its implementation and efficiency [7-10]. The dynamics equations in a recursive formulation are written in terms of the system's degree of freedom and, typically, a lower dimensionality than those in an augmented formulation.

As shown in Figure 1, the bodies are connected with joints. Each body number can be represented as B$_n$ and the preceding body could be named B$_{n-1}$. The arranged set of body numbers is called "body connection array" [11]. The joint number is named before the body number. Each joint can consist of a single axis or multiple axes such as revolute, prismatic, cylindrical, translational and spherical joints.

2.1.1 Kinematics

Relative motion between neighboring bodies and constraints are the two main aspects in recursive kinematics to generate the total system matrices and a solution for the equation of motion for the multibody system. Figure 2 shows the elementary system of two bodies interconnected by a joint.

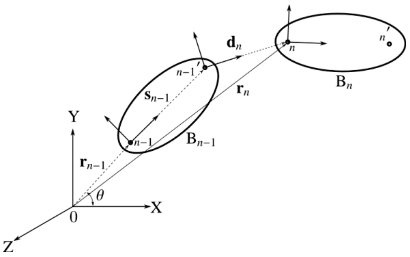


Figure 2. Relationship of contiguous bodies

The orientation relationship of contiguous bodies is obtained by sequentially transforming from the body reference frame on body B$_0$ to the body reference frame on body B$_n$. The system of bodies is considered an open chain and point 0 is the global reference frame. The relative kinematics of position for each body can be described with respect to the global reference as

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{s}_{n-1} + \mathbf{d}_n \tag{1}$$

where $\mathbf{d}_n$ is joint relative displacement vector from point n-1' to n while $\mathbf{s}_{n-1}$ can be computed as

$$\mathbf{s}_{n-1} = \mathbf{A}_{n-1}\overline{\mathbf{s}}_{n-1} \tag{2}$$

Orthogonal rotation matrix $\mathbf{A}_{n-1}$ is obtained from previous body B$_{n-1}$. As an example in this case, the rotation matrix $\mathbf{A}_{n-1}$ at z-axis can be expressed as

$$\mathbf{A}_{n-1} = \underbrace{\begin{bmatrix} \cos(\theta_{n-1}) & -\sin(\theta_{n-1}) & 0 \\ \sin(\theta_{n-1}) & \cos(\theta_{n-1}) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{z-axis} \tag{3}$$

where the rotation matrix $\mathbf{A}_n$ of body n can be computed as

$$\mathbf{A}_n = \mathbf{A}_{n-1}\mathbf{A}_d \tag{4}$$

where $\mathbf{A}_d$ is the relative rotation matrix between contiguous bodies. The velocities could be determined from the time derivative of equation (1).

$$\begin{aligned} \mathbf{v}_n &= \dot{\mathbf{r}}_n \\ &= \dot{\mathbf{r}}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} + \dot{\mathbf{d}}_n \end{aligned} \tag{5}$$

$$\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} + \boldsymbol{\omega}_d \tag{6}$$

Matrix $\tilde{\boldsymbol{\omega}}_{n-1}$ is the skew-symmetric matrix of cross product $\tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} \equiv \boldsymbol{\omega}_{n-1} \times \mathbf{s}_{n-1}$. Vector $\boldsymbol{\omega}_d$ is the relative angular velocity of the joint. The acceleration equations can be obtained by differentiating the equation (5) and (6) as follow

$$\ddot{\mathbf{r}}_n = \ddot{\mathbf{r}}_{n-1} + \dot{\tilde{\boldsymbol{\omega}}}_{n-1}\mathbf{s}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1}\tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} + \ddot{\mathbf{d}}_n \tag{7}$$

$$\dot{\boldsymbol{\omega}}_n = \dot{\boldsymbol{\omega}}_{n-1} + \dot{\boldsymbol{\omega}}_d \tag{8}$$

In the equations (1)-(8), the terms $\mathbf{d}_{\mathrm{n}}$, $\mathbf{A}_d$ and $\boldsymbol{\omega}_d$ are the function of relative joint position, joint rotation matrix and joint angular velocity, respectively. The functions of velocity and acceleration for each common type of joint (prismatic, spherical etc.) can be obtained from the derivation of the expressions as suggested in [12].

2.1.2 Equation of motion

The equation of motion can be developed from the principle of virtual power as introduced in [13]. In Figure 4, assuming that position vector $\mathbf{r}_n$ is shifting to the centre of gravity of the body and becomes $\mathbf{u}_n$, $\boldsymbol{\omega}_n$ is the velocity of body n and N_b is the number of contiguous bodies. The equation of motion can be written in matrix form as

$$\sum_{n=1}^{N_b}\left\{\delta\dot{\mathbf{u}}_n^{\mathrm{T}}\delta\boldsymbol{\omega}_n^{\mathrm{T}}\right\}\left(\begin{bmatrix}\mathbf{m}_n\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_n\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{u}}_n\\ \dot{\boldsymbol{\omega}}_n\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\ \tilde{\boldsymbol{\omega}}_n\mathbf{J}_n\boldsymbol{\omega}_n\end{bmatrix}-\begin{bmatrix}\mathbf{F}_n\\ \mathbf{T}_n\end{bmatrix}\right)=0 \tag{9}$$

where $\delta\dot{\mathbf{r}}_n$ and $\delta\boldsymbol{\omega}_n$ are virtual translational and angular velocities, matrix $\mathbf{J}_n$ is inertia which is obtained from $\mathbf{A}_n\mathbf{J}_n\mathbf{A}_n^{\mathrm{T}}$, $\mathbf{m}_n$ is mass of the body while $\mathbf{F}_n$ and $\mathbf{T}_n$ are external forces and torques. The equation of motion can be formed into simple form as

$$\sum_{n=1}^{N_b}\delta\dot{\mathbf{q}}_n^{\mathrm{T}}\left(\mathbf{M}_n\ddot{\mathbf{q}}_n+\mathbf{C}_n-\mathbf{Q}_n\right)=0 \tag{10}$$

or in the compact form as

$$\delta\dot{\mathbf{q}}^{\mathrm{T}}\left(\mathbf{M}\ddot{\mathbf{q}}+\mathbf{C}-\mathbf{Q}\right)=0 \tag{11}$$

Where $\dot{\mathbf{q}}^{\mathrm{T}}$, $\mathbf{C}^{\mathrm{T}}$ and $\mathbf{Q}^{\mathrm{T}}$ are vectors with dimension N=6N_b while matrix **M** is a diagonal mass matrix.

From the equation (11), the dependent virtual velocities $\delta\dot{\mathbf{q}}$ should be eliminated in order to obtain the final equation with a small size of dimension. Therefore, the independent virtual velocities $\delta\dot{\mathbf{z}}$ vector with N_f number of degree of freedom, which bring out from the joint velocities, is introduced. As mentioned before, $\dot{\mathbf{q}}$ have a dimension of 6N_b which covers the translational and angular velocities of each body. However, the angular position cannot be obtained by integrating $\dot{\mathbf{q}}$ which necessitates the introduction of the new variable $\mathbf{p}_q$ to represent the dependent position. It can be done by separately defining the translational and orientation as suggested in the Cartesian coordinate approach [14] or other suitable methods. In this case, $\dot{\mathbf{q}}$ and $\dot{\mathbf{p}}_q$ can be represent as

$$\dot{\mathbf{p}}_q=\mathbf{E}_q\dot{\mathbf{q}} \tag{12}$$

$$\dot{\mathbf{q}}=\mathbf{F}_q\dot{\mathbf{p}}_q \tag{13}$$

Where $\mathbf{E}_q$ is velocity transformation matrix and $\mathbf{F}_q$ is inverse velocity transformation matrix. Like $\dot{\mathbf{q}}$, $\dot{\mathbf{z}}$ could also be integrated to obtain position which necessitates introducing position vector $\mathbf{p}_z$ and velocity transformation matrix $\mathbf{E}_z$ with size of (N_f x N_f).

$$\dot{\mathbf{p}}_z=\mathbf{E}_z\dot{\mathbf{z}} \tag{14}$$

In this particular equation, $\mathbf{p}_q$ can be represented in the function of $\mathbf{p}_z$ and time as

$$\mathbf{p}_q=\mathbf{p}_q(\mathbf{p}_z,t) \tag{15}$$

By taking derivative of equation (15), the final $\dot{\mathbf{q}}$ can be obtained as

$$\begin{aligned}\dot{\mathbf{p}}_q &= \frac{\partial \mathbf{p}_q}{\partial \mathbf{p}_z}\dot{\mathbf{p}}_z + \frac{\partial \mathbf{p}_q}{\partial t} \\ &= \frac{\partial \mathbf{p}_q}{\partial \mathbf{p}_z}\mathbf{E}_z\dot{\mathbf{z}} + \frac{\partial \mathbf{p}_q}{\partial t}\end{aligned} \tag{16}$$

Replacing $\dot{\mathbf{p}}_q$ with equation (13)

$$\frac{\dot{\mathbf{q}}}{\mathbf{F}_q} = \frac{\partial \mathbf{p}_q}{\partial \mathbf{p}_z}\mathbf{E}_z\dot{\mathbf{z}} + \frac{\partial \mathbf{p}_q}{\partial t} \tag{17}$$

$$\dot{\mathbf{q}} = \mathbf{F}_q\frac{\partial \mathbf{p}_q}{\partial \mathbf{p}_z}\mathbf{E}_z\dot{\mathbf{z}} + \mathbf{F}_q\frac{\partial \mathbf{p}_q}{\partial t} \tag{18}$$

The final velocities equation (18) can be written in the simple form as

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} + \mathbf{b} \tag{19}$$

From here, with the assumption that $\mathbf{p}_q$ is not time dependent, $\frac{\partial \mathbf{p}_q}{\partial t}$ can be eliminated and virtual velocities can be written as

$$\delta\dot{\mathbf{q}} = \mathbf{R}\delta\dot{\mathbf{z}} \tag{20}$$

By taking time derivative of equation (19), the acceleration equation is

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}} \tag{21}$$

Substituting equations (20) and (21) into the original equation (11) could lead to

$$\begin{aligned}\mathbf{R}^{\mathrm{T}}\delta\dot{\mathbf{z}}^{\mathrm{T}}\left(\mathbf{M}(\mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}}) + \mathbf{C} - \mathbf{Q}\right) &= 0 \\ \mathbf{R}^{\mathrm{T}}\delta\dot{\mathbf{z}}^{\mathrm{T}}(\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} + \mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{M}\dot{\mathbf{b}} + \mathbf{C} - \mathbf{Q}) &= 0\end{aligned} \tag{22}$$

$$\delta\dot{\mathbf{z}}^{\mathrm{T}}(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{b}} + \mathbf{R}^{\mathrm{T}}\mathbf{C} - \mathbf{R}^{\mathrm{T}}\mathbf{Q}) = 0 \tag{23}$$

From equation (23), virtual velocity $\delta\dot{\mathbf{z}}$ can be eliminated which leads to final equation as

$$\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{b}} + \mathbf{R}^{\mathrm{T}}\mathbf{C} - \mathbf{R}^{\mathrm{T}}\mathbf{Q} = 0 \tag{24}$$

Equation (24) can be reconstructed into new simplified form as

$$\begin{aligned}\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} &= -\mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} - \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{b}} - \mathbf{R}^{\mathrm{T}}\mathbf{C} + \mathbf{R}^{\mathrm{T}}\mathbf{Q} \\ &= \mathbf{R}^{\mathrm{T}}(\mathbf{Q} - \mathbf{C}) - \mathbf{R}^{\mathrm{T}}\mathbf{M}(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}})\end{aligned} \tag{25}$$

2.1.3 Computing matrix R

Here, it is clearly shown that velocity transformation matrix **R** plays an important role in the equation. Due to that, it is important to give attention to solving the matrix **R** which affects each body. As an example referring to the simple analogy in Figure 5, the total size of matrix **R** by assuming that all joints have one d.o.f is (30x5) where the columns of matrix **R** correspond to each joint in the system. If body B$_3$ has been chosen to identify the matrix $\mathbf{R}_3$, the column corresponding to joints 4 and 5 are zero and only joints 1, 2 and 3 needs to be computed. Therefore, matrix $\mathbf{R}_3$ is a matrix of $\left[\mathbf{R}_3^1\mathbf{R}_3^2\mathbf{R}_3^3\right]$ with size of (6x3) matrix where superscripts represent the joint number. Each sub matrix is (6x1) vector and the first three components of the vector obtain from equation (5) and last three components from equation

(6). From here it can be generalized that $\mathbf{R}_n$ can be formed into $\left[\mathbf{R}_n^1\mathbf{R}_n^2...\mathbf{R}_n^{N_q}\right]$ with the size of (6xN_d) where N_d and N_q is the number of d.o.f and number of joints from body n to the vehicle body. The size of the sub-matrix $\mathbf{R}_n^j$ is (6xN_j) where N_j is the number d.o.f at joint j.

2.1.4 Numerical algorithm for open loop system

The system of differential algebraic equation (DAE) has been transformed into ordinary differential equation (ODE) by integrating variables $\ddot{\mathbf{z}}$ and $\dot{\mathbf{p}}_z$. In general, the algorithm is derived for generating the real-time simulation for an open loop system (chain topology) as in Figure 3.

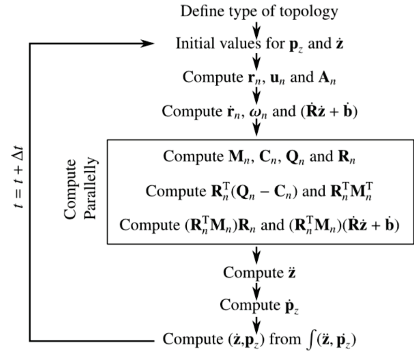

Define type of topology
Initial values for $\mathbf{p}_z$ and $\dot{\mathbf{z}}$
Compute $\mathbf{r}_n$, $\mathbf{u}_n$ and $\mathbf{A}_n$
Compute $\dot{\mathbf{r}}_n$, ω_n and $(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}})$
Compute $\mathbf{M}_n$, $\mathbf{C}_n$, $\mathbf{Q}_n$ and $\mathbf{R}_n$
Compute $\mathbf{R}_n^{\mathrm{T}}(\mathbf{Q}_n - \mathbf{C}_n)$ and $\mathbf{R}_n^{\mathrm{T}}\mathbf{M}_n^{\mathrm{T}}$
Compute $(\mathbf{R}_n^{\mathrm{T}}\mathbf{M}_n)\mathbf{R}_n$ and $(\mathbf{R}_n^{\mathrm{T}}\mathbf{M}_n)(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}})$
Compute Parallelly
$t = t + \Delta t$
Compute $\ddot{\mathbf{z}}$
Compute $\dot{\mathbf{p}}_z$
Compute $(\dot{\mathbf{z}},\mathbf{p}_z)$ from $\int(\ddot{\mathbf{z}}, \dot{\mathbf{p}}_z)$

Figure 3. Algorithm for open loop real-time simulation

2.2 Modeling of actuators

In this study, the tree harvester systems are assumed to be driven with hydraulic actuators. Hydraulic actuators can be modelled using the lumped fluid theory. The lumped fluid theory divides a hydraulic circuit into volumes wherein the pressure is assumed to be equally distributed. The valves are modelled employing a semi-empirical approach which allows obtaining the parameters used in the flow equations through the orifices, in many cases from manufacturer catalogues. Usually, the hydraulic system has high nominal frequencies, and for this reason, the time step must be short in order to produce reliable results. Since one time integration algorithm is used for the machine system, the time integration in hydraulics forces the time step in the integration of mechanical components to be small. In this approach, the pressure waves in pipes and hoses are assumed to be insignificant [15]. The hydraulic pressure in each hydraulic volume i can be described as

$$\dot{p}_l = \frac{B_{ei}}{V_i}\sum_{j=1}^{n_c} Q_{ij} \tag{26}$$

where B_{ei} is the effective bulk modulus of volume i, equation (21), Q_{ij} is the outgoing or incoming flow rate of volume i and n_c is the total number of flows related to volume i. The effective bulk modulus defines the flexibility of the hydraulics and it can be calculated as

$$B_{ei} = \frac{1}{\dfrac{1}{B_{oil}} + \displaystyle\sum_{j=1}^{nc} \frac{V_j}{V_i B_j} + \sum_{k=1}^{n_h} Q_{ij} \frac{V_k}{V_i B_k}} \tag{27}$$

In equation (22), n_h is the total number of pipes and hoses related to volume i. The bulk modulus of oil, B_{oil}, accounts for the amount of non-dissolving air in oil and it is a function of pressure. The maximum value is typically B_{oilmax} = 1.6e9Pa. It is important to note that the bulk modulus B_j of component j is also dependent on the component type.

2.2.1 Modeling of Valves

For modelling purposes, a valve is assumed to consist of several adjustable restrictor valves which can each be modelled separately [16]. With small pressure differences (pressure difference < 1 bar), the flow over the restrictor is thought to be laminar, whereas with larger differences, it is thought to be turbulent. When using the semi-empiric modelling method, the flow over the restrictor can be written as

$$Q = C_v U \sqrt{dp} \tag{28}$$

where C_v is the semi-empiric flow rate constant and defines the size of the valve and U is a variable that defines the spool or poppet position. For a number of valve types, the variable U can be defined using a first order differential equation as

$$\dot{U} = \frac{U_{ref} - U}{\tau} \tag{29}$$

where U_{ref} is the spool reference position and t time constant describing the dynamics of the valve spool.

2.2.2 Modeling of Hydraulic Cylinders

A hydraulic cylinder can be modelled by simply using the dimensions of the cylinder and the pressure obtained from equation (21). The motion of the hydraulic cylinder produces a flow rate to the hydraulic volume I as follows

$$\begin{aligned} Q_{j1} &= -\dot{x}A_1 \\ Q_{j2} &= \dot{x}A_2 \end{aligned} \tag{30}$$

where A_1 is the area of cylinder piston side and A_2 is the area on cylinder piston rod side, $\dot{x}$ is the velocity of the stroke of the cylinder, as depicted in Figure 4.

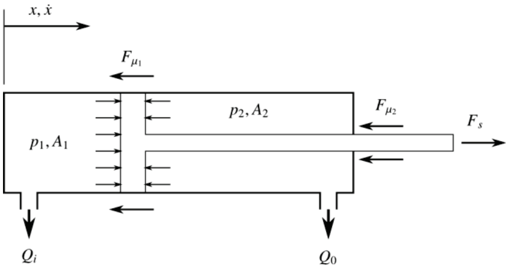


Figure 4. Hydraulic cylinder transforms hydraulic pressure into a mechanical force

The force produced by the hydraulic cylinder can be defined as

$$F_s = p_1 A_1 - p_2 A_2 - F_\mu \tag{31}$$

where F_μ is the total friction force of the cylinder and p_1 and p_2 are pressures acting in the cylinder chambers. The friction force is a function of pressures, cylinder efficiency, η and velocity. The friction force can be described in a simple case as follow:

$$F_\mu = (p_1 A_1 - p_2 A_2)(1-\eta) f(\dot{x}) \tag{32}$$

The velocity dependent co-efficient, $f(\dot{x})$ can be described using a spline-curve as shown in Figure 5.

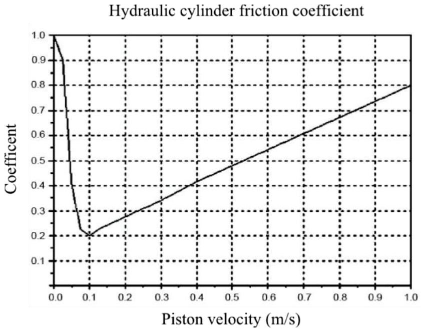


Figure 5. The velocity dependent friction co-efficient

2.3 Collision and contact modeling

Contact model is one of the important models that should be considered when simulating more than one body. It's function is to avoid the interpenetration of two bodies. Contact algorithm is used to determine when and where contact occurs and to calculate the elements of contact. Contact analysis can be done according to two numerical approaches, namely finite element analysis (FEM) and multibody system (MBS). FEM is the most powerful tool to analyze contact, and the results are very precise compared to MBS, but the computing time it requires is too long. Therefore, to reduce the computing time costs and come closer to real-time simulation, MBS is the best choice with acceptably accurate results.

In practice, there are two steps in modeling the contact: contact detection and collision response [17]. Contact detection is the first and most important step. Contact detection models the moving bodies and determines when and where a potential collision will occur. The accuracy of the results depends on the geometry of the contacting bodies. Contact response prevents the penetration between two bodies. This is done by analyzing the contact force of colliding bodies based on their geometric properties, relative velocities and material. However, the penetration between two bodies may occur at high speed at a single time-step especially when using a bigger time step [18].

There are many formulations for treating the contact in multibody systems such as the penalty technique [19], the Lagrange multiplier method [20], the impulse method [21], the polygonal contact model [22] and the time stepping method [23]. In this study, the penalty method is used to model the contact in multibody systems. Penalty method is one of the well-known approaches. It is also known as surface compliance or soft contact, and it permits interpenetration [24]. The general algorithm for contact/collision models can be seen in the state transition diagram in Figure 6.

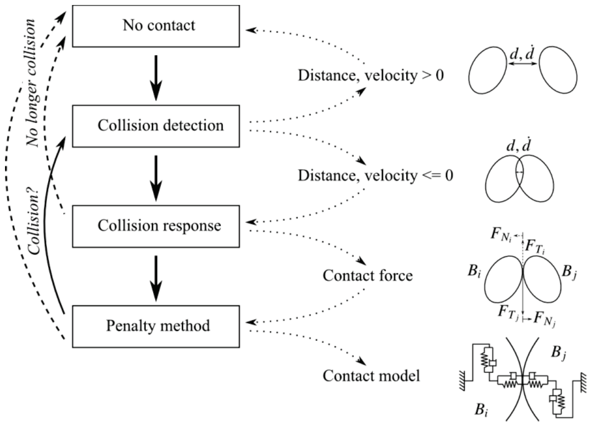


Figure 6. State transition diagram track of contact phase between two bodies

2.3.1 Collision detection

In this study, the collision between bodies with different geometry is detected using a well-known approach called bounding volume (BV). This method is efficient enough to estimate whether the objects are overlapping or not in collision with detected trees. BV uses simple bounding volumes such as spheres and boxes which encapsulate the more complex geometry of the object. By using these simple shapes, it can accelerate the collision detection process. There are a few types of bounding methods which can be used for collision detection such as bounding spheres [25], Axis-Aligned Bounding Box (AABB) [26], Discrete-Orientation Polytope (k-DOP) [27] and Oriented Bounding Box (OBB) [28].

For example, in the AABB method, the collision can be estimated by comparing the distance between centre position, c_n and radii $r^x{}_n$ and $r^y{}_n$ of the bounding boxes. Figure 7 shows a simple example of the AABB method used for harvester head and log.

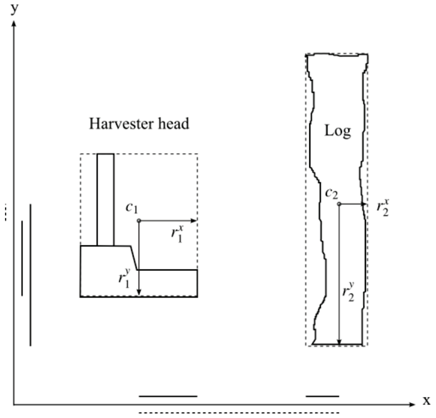


Figure 7. Axis Aligned Bounding Box (AABB)

In general, a collision will not occur if,

$$\left|(c_1 - c_2)\begin{pmatrix}1\\0\end{pmatrix}\right| > r_1^x + r_2^x$$
$$\left|(c_1 - c_2)\begin{pmatrix}0\\1\end{pmatrix}\right| > r_1^y + r_2^y \qquad (33)$$

Therefore, it can be said that selecting the suitable method for the bounding volume method has to be based on the type of objects. A different method may give different results and affect the after collision result.

In certain conditions, collision detection needs special treatment, for example, when dealing with particles. Even though this method will affect the computing cost especially when a large amount of particles is involved, it will produce good results to be used in extended simulation. Particle based collision detection in general can be divided into several steps such as neighbor search method and force model as explained in [29].

2.3.2 Collision response

When a collision occurs, the existing simulation model is stopped and a new simulation model with new velocities as the new initial condition will take over as after collision model. However, it is difficult to avoid penetration between the objects, even using smaller time-step in the collision prediction algorithm. Smaller time-step could reduce the penetration amount but increase the computing time. This could result in an inaccurate simulation and produce errors in the simulation system. Therefore, the penalty method which is known as 'soft contact' is introduced to minimize this error. This method can almost fulfill the exact contact conditions and its accuracy depends highly on the penalty parameter and the specific case. In this method, the contact force can be determined by applying a non-linear spring-damper element at the contact point. This spring-damper element can be described in detail using Hertz impact [30,31].

2.3.2.1 Kinematics

The kinematics of the collision point can be described using a global reference frame. Figure 8 shows the example of collision direction before and during the collision.

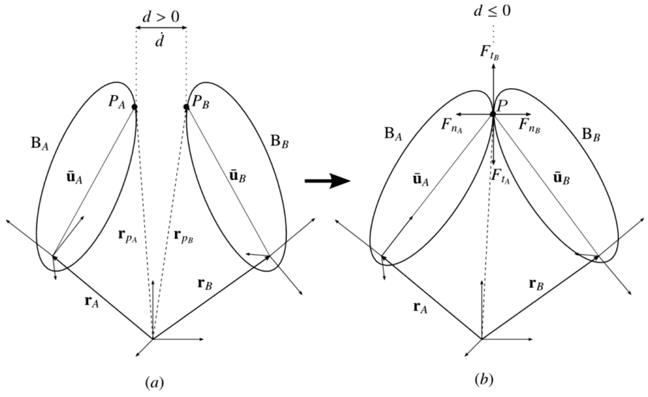


Figure 8. Collision between bodies A and B.

The distance, *d* can be written from the basic global reference frame approach as follows:

$$\mathbf{r}_{P_A} = \mathbf{r}_A + \mathbf{A}_A\bar{\mathbf{u}}_A$$
$$\mathbf{r}_{P_B} = \mathbf{r}_B + \mathbf{A}_B\bar{\mathbf{u}}_B \qquad (34)$$

$$\mathbf{d} = \mathbf{r}_{P_B} - \mathbf{r}_{P_A} = \mathbf{r}_B + \mathbf{A}_B\bar{\mathbf{u}}_B - \mathbf{r}_A - \mathbf{A}_A\bar{\mathbf{u}}_A \qquad (35)$$

where $\mathbf{A}_A$ and $\mathbf{A}_B$ are a rotation matrix of bodies *A* and *B* in three dimensional and $\bar{\mathbf{u}} = [x\ y\ z]^T$ is relative position vector of bodies *A* and *B*. Knowing that, the relative velocity is the time derivative of *d*, therefore the relative velocity is,

$$\begin{aligned}\dot{\mathbf{d}} &= \dot{\mathbf{r}}_B + \tilde{\boldsymbol{\omega}}_B \bar{\mathbf{u}}_B + \mathbf{A}_B \dot{\bar{\mathbf{u}}}_B - \dot{\mathbf{r}}_A - \tilde{\boldsymbol{\omega}}_A \bar{\mathbf{u}}_A - \mathbf{A}_A \dot{\bar{\mathbf{u}}}_A \\ &= \dot{\mathbf{r}}_B - \tilde{\bar{\mathbf{u}}}_B \boldsymbol{\omega}_B + \mathbf{H}_B \dot{\mathbf{q}}_B - \dot{\mathbf{r}}_A + \tilde{\bar{\mathbf{u}}}_A \boldsymbol{\omega}_A - \mathbf{H}_A \dot{\mathbf{q}}_A\end{aligned} \tag{36}$$

Where

$$\mathbf{H} = \mathbf{A}\left[\frac{d\bar{\mathbf{u}}}{dt}\right]$$

$$\dot{\mathbf{q}} = \dot{\boldsymbol{\theta}}$$

$$\tilde{\bar{\mathbf{u}}} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

2.3.2.2 Contact forces

As mentioned earlier, the contact forces apply Penalty method where small penetration is allowed between the contacting bodies. Basically, contact force, F_c can be composed from the tangential and normal forces components as,

$$F_c = F_t + F_n \tag{37}$$

The tangential (F_t) and normal (F_n) forces can be approximated as

$$F_n = F_{ni}(\delta_{ni}) + g_{ni}(\delta_{ni}, \dot{\delta}_{ni}) \tag{38}$$

$$F_t = F_{ti}(\delta_{ti}) + g_{ti}(\delta_{ti}, \dot{\delta}_{ti}) \tag{39}$$

Where f_{ni} and f_{ti} are the elastic stiffness term and, g_{ni} and g_{ti} are the damping term. The contact force procedure for this study has been described in [23].

3 Numerical Example

The numerical example is based on a Tree harvester truck as shown in Figure 9. The simulation model of the harvester consists of a number of bodies and it is modeled using recursive method. Figure 10 shows the simplified topology of the harvester which is used as a reference while implementing the recursive method. The models include several force components such as boom, hydraulics and log cutter (harvester head).

A mechanics of a tree harvester truck can be considered as a set of rigid bodies, while mechanical joints will limit the relative motion of coupled bodies. The configuration of a mechanism changes in time based on the forces and motions applied to its components. Kinematic models concern the motion of the system independently without taking into account the forces that produce the motion. The kinematic model will show how the bodies move in the system and it needs linear and nonlinear systems of equations to solve. However, to achieve realistic responses in regard to the motion of the system that is caused by applied force, a dynamic model must be used. To develop the kinematic and dynamic model, the structure of the machine needs to be identified at an early stage to ensure a steady structure will be used for all the procedures.

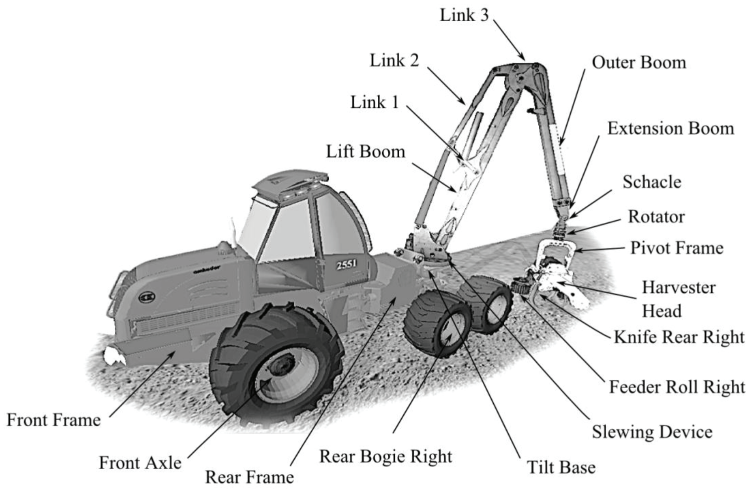


Figure 9. Tree harvester truck connecting components

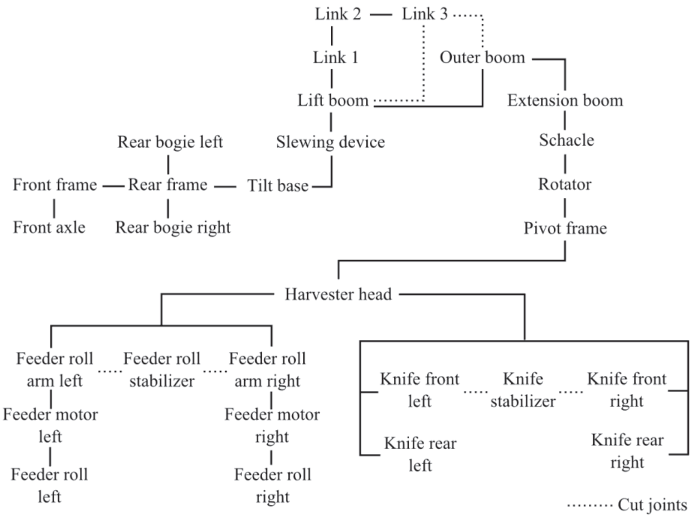


Figure 10. Topology of tree harvester truck model.

Since the model under investigation is used in training simulators, the hydraulic model example for this study is a simplified version of the actual hydraulics of outer boom as shown in Figure 11.

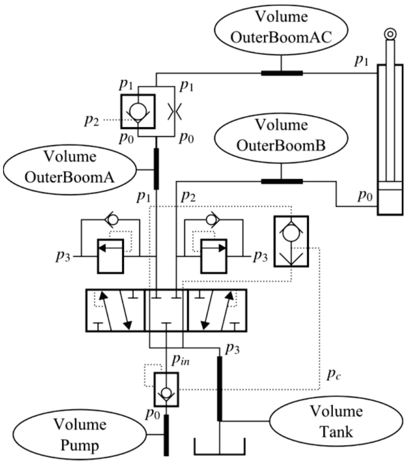


Figure 11. Simplified description of outer boom hydraulic system.

Hydraulic circuits consist of hydraulic cylinder, pressure compensated proportional directional valves, pressure relief valves and pumps. Figures 12 and 13 show examples of a simple work cycle. In Figure 12, the valve control signal and spool opening are presented. Figure 13 presents the flow rates through the outer boom circuit valve, and the pressure rates of cylinder chambers are shown in Figure 14.

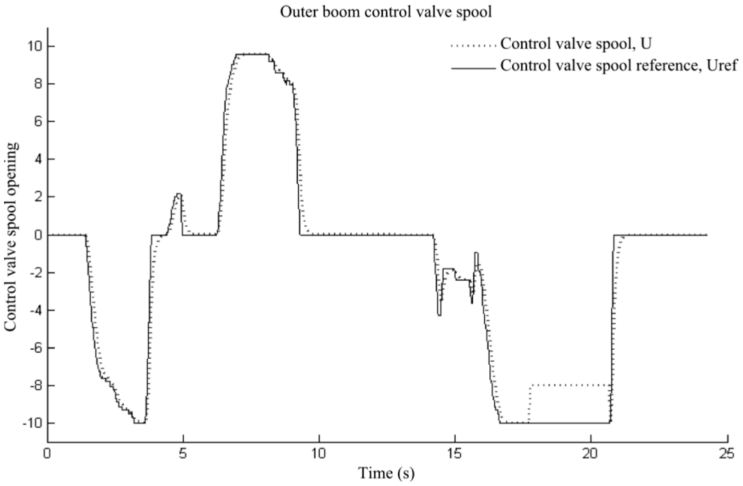


Figure 12. Control valve signal and reference control signal.

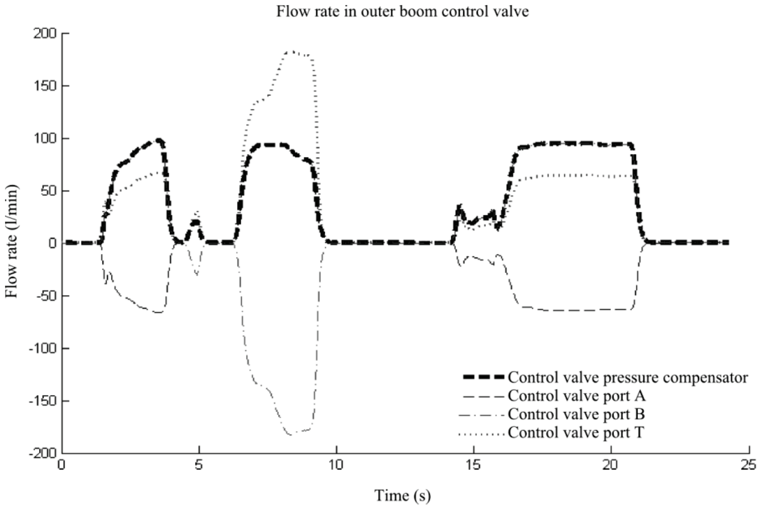


Figure 13. Flow rates through the valve during work cycle.

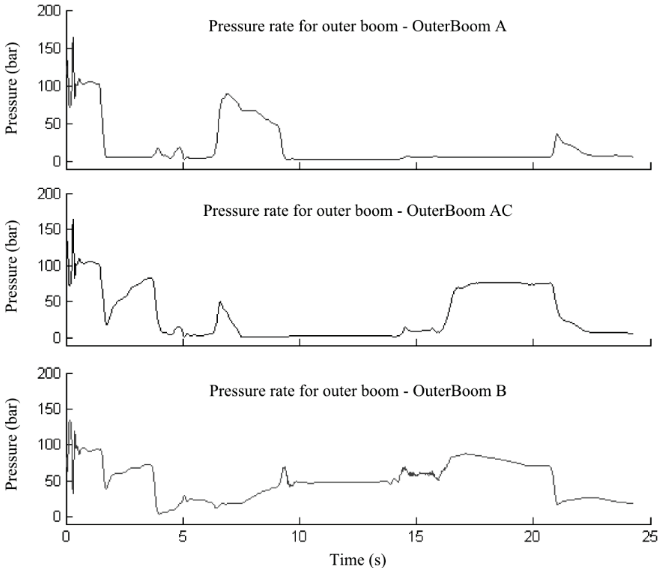


Figure 14. Pressure rates at the inlet (A and AC) and outlet (B) of the cylinder.

The simulation of the system was modeled using MeVEA Real-Time simulation environment. The environment offers the possibility of off-line simulation for a more detailed model or, alternatively, real-time simulation and visualization for simplified models [32]. The environment is compatible with MeVEA Full Mission Solution, which offers a motion platform and visualization environment combined with user interface – including an operators seat, joysticks and pedal etc. based on the application [33].

4 Conclusion

In real-time simulation, a machine needs to be considered as a coupled system that consists of mechanical components and actuators. This study introduced a general simulation approach that can be applied to real-time simulation of a hydraulically driven harvester. The introduced approach was based on the use of the recursive method that was coupled with the lumped fluid theory for the modelling of hydraulic circuits and the penalty method for contact modelling. The introduced simulation approach was applied to create real-time simulation models of a tree harvester truck. The simulation model of the harvester was embedded into real-time simulation environment that consists of visualization, motion platform and an I/O interface. The real-time simulation environment allows for merging a user to the simulation model in a realistic manner.

References

[1] P.M. Menghal and A. Jaya Iaxmi. Real time simulation: A novel approach in engineering education. *IEEE*, 2011.

[2] E. Papadopoulos. On the modeling and control of an experimental harvester machine manipulator. *Proceeding IROS 97. IEEE*, 1997.

[3] P. Fisette, O.Bruls and J. Swevers. Multiphysics modeling of mechatronic multibody systems. *Proceeding of International Conference on Noise and Vibration Engineering (ISMA2006)*, page 41-67, 2006.

[4] J.C. Samin, O. Bruls, J.F. Collard, L. Sass and P. Fisette. Multiphysics modeling and optimization of mechatronic multibody systems. *Multibody System Dynamic*, 18: 345-373, 2007.

[5] D.S. Bae and E.J. Haug. A recursive formulation for constrained mechanical system dynamics: Part 1. Open Loop System. *Mechanics of Structure and Machines*, 15(3):359-382, 1987.

[6] W. Hooker and G. Margulies. The dynamical attitude equation for an n-body satellite. *Journal of Astronautical Science,* 12:123-128, 1965.

[7] R. Featherstone. The calculation of robot dynamics using articulated-body inertias*. International Journal of Robotics Research,* 2:13-30, 1983.

[8] D.S. Bae and E.J. Haug. A recursive formulation for constrained mechanical system dynamic: Part II. Closed loop system*. Mechanics of Structure and Machines,* 15(4):481-506, 1988.

[9] S.S. Kim and E.J. Haug. A recursive formulation for flexible multibody dynamics. Part I: Open loop system. *Computer Methods in Applied Mechanics and Engineering,* 71:481-506, 1988.

[10] D.S. Bae, J.M. Han and H.H. Yoo. A generalized recursive formulation for constrained mechanical system dynamic. *Mechanics of Structure and Machines,* 27(3):293-315, 1999.

[11] P.M.A. Slaats. Recursive Formulations in Multibody Dynamics. *Computational Mechanics*. 1991.

[12] A. Avello, J.M. Jimenez, E. Bayo and J. Garcia de Jalon. A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformation. *Computer Methods in Applied Mechanics and Engineering 107, North Holland,* pages 313-339, 1993.

[13] E.J. Haug. Computer aided kinematics and dynamics of mechanical systems, Volume I, Basic Methods. *Allyn and Bacon*, Boston, MA, 1989.

[14] J. Garcia de Jalon and E. Bayo. Kinematic and dynamic simulation of multibody systems, The real-time challenge. *Springer-Verlag*, New York, 1994.

[15] O. Dmitrochenko. Efficient methods of numerical simulation of dynamics of nonlinear rigid and flexible multibody systems, Ph.D. (Cand. of Phys. and Math. Sc.) *Dissertation in the field of Theoretical Mechanics,* Moscow State University (Russia), 2004.

[16] J. Watton. Fluid Power Systems. *Prentice Hall International (UK) Ltd.* 1989.

[17] H.M. Handroos and M.J. Vilenius. Flexible Semi-Empirical Models for Hydraulic Flow Control Valves. *Journal of Mechanical Design,* 113(3):232–238, 1991.

[18] S. Ebrahimi and P. Eberhard. Aspects of Contact Problems in Computational Multibody Dynamics. *Multibody Dynamics: Computational Methods and Applications*, Springer, 2007.

[19] S. Moisio, P.M. Korkealaakso and A.J. Rouvinen. Contact Modeling in Real-Time Simulation of an Underground Wheeled Loader. *The 1st Joint International Conference on Multibody System Dynamics*. Lappeenranta, Finland, 2010.

[20] E. Drumwright. A Fast and Stable Penalty Method for Rigid Body Simulation. *IEEE Transaction Visualization and Computer Graphics*, 14(1):231-240, 2008.

[21] K.E. Zaazaa and A.A. Shabana. A Technique for Validating a Multibody Wheel/Rail Contact Algorithm. Proceeding of DETC'03. *ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Chicago, USA, 2003.

[22] S. Lembcke. Realtime Rigid Body Simulation Using Impluses. *UMM CSCI Senior Seminar Conference,* Morris Minnesota, 2006.

[23] S. Ebrahimi, G. Hippmann and P. Eberhard. Extension of the Polygonal Contact Model for Flexible Multibody Systems. *International Journal of Applied Mathematics and Mechanics*, Vol. 1, pp. 33-50. 2005.

[24] M. Anitescu. A Fixed Time-Step Approach for Multibody Dynamic with Contact and Friction. *International Conference on Intelligent Robots and System*, 2003.

[25] E. Drumwright and D.A. Sheel. An Evaluation of Methods for Modeling Contact in Multibody Simulation. *IEEE International Conference on Robotics and Automation. Shanghai International Conference Center*, Shanghai, China, 2011.

[26] S. Redon, A. Kheddary and S. Coquillart. Fast continuous collision detection between rigid bodies. *Computer Graphics Forum,* 21(3):279–288, 2002.

[27] K.C. Lai and S.C. Kang. Collision detection strategies for virtual construction simulation. *Automation in Construction*, 18(6):724-736, 2009.

[28] G. Zachmann. Rapid collision detection by dynamically aligned DOP-trees. *Proceedings of the IEEE virtual reality annual international symposium*, pages 90–97, 1998.

[29] S. Gottschalk, M.C. Lin and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer graphics (SIGGRAPH '96 proceedings*), pages 171–80, 1996.

[30] B. Muth, M.K. Muller, P. Eberhard and S. Luding. Collision Detection and Administration Methods for Many Particles with Different Sizes. *4th International Conference on Discrete Element Methods, DEM 2007*, Brisbane, Australia, 2007.

[31] K.L. Johnson. Contact Mechanics, *Cambridge University Press*, 1985.

[32] A.J. Rouvinen, T. Lehtinen and P.M. Korkealaakso. Container gantry crane simulator for operator training. *Proceeding IMechE Vol.219. Part K: Journal of Multibody Dynamics*, 2005.

[33] P.M. Korkealaakso, A.J. Rouvinen, S.M. Moisio and J.K. Peusaari. Development of real-time simulation environment. *Multibody System Dynamic,* 17:177-194, 2007.

Publication III

Baharudin, E., Rouvinen, A., Korkealaakso, P., and Mikkola, A.
Real-time Multibody Application for Tree Harvester Truck Simulator

Reprinted with permission from
Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics
DOI: 10.1177/1464419314521182

Original Article

Proc IMechE Part K:
J Multi-body Dynamics
0(0) 1–17
© IMechE 2014
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1464419314521182
pik.sagepub.com

Real-time multibody application for tree harvester truck simulator

Mohamad Ezral Baharudin[1], Asko Rouvinen[2],
Pasi Korkealaakso[2] and Aki Mikkola[1]

Abstract

A real-time simulator for a tree harvester has been developed for training in more effective vehicle and cutter operation and tree management. The equations of motion of the constrained mechanical system of the tree harvester are expressed using a recursive formulation. The hydraulic actuator modelling of the harvester is based on lumped fluid theory, in which the hydraulic circuit is divided into discrete volumes where pressures are assumed to be distributed equally, while pressure wave propagation in pipes and hoses is assumed to be negligible. For modelling purposes, valves are broken up into a number of adjustable restrictors, which can be modelled separately. The contact model used comprises two parts: collision detection and response. Collision detection identifies whether, when and where moving bodies may come in contact. Collision response prevents penetration when contact occurs and identifies how it should behave after collision. A penalty method is used in this study to establish object collision events. The major achievement of this study is combining these three modelling methods in the application of a real-time simulator.

Keywords

Multibody dynamics, recursive method, hydraulic actuator, harvester truck, contact modelling

Date received: 27 August 2013; accepted: 3 January 2014

Introduction

As computer simulation tools become more sophisticated, the option of virtually building and analysing complex machine systems becomes an increasingly attractive alternative to physical prototyping. If the simulation is used for operator training, the simulation system should be able to present the working environment and machine functions accurately to give the operator a true-to-life experience. A number of tree harvester training simulators have been developed and are already being marketed by well-known companies. These include, for example, the John Deere Forestry Machine Simulator, the Valmet Komatsu Forestry Simulator and the Creanex Training Simulator.

In the past, tree harvester simulators required a great deal of space and highly skilled (expensive) teams to manage set-up and maintenance.[1] However, with the increasing sophistication of computer technology and growing knowledge of multibody systems, machine simulators are becoming simpler and more user friendly. In practice, common real-time machine simulation should consist of the description of several subsystems such as the machine mechanics, the actuators and the control system. The development of a comprehensive computer-based tree harvester simulator is claimed to have started in the year 1997. The simulation models involved two major sub-models: the kinematic of four degrees of freedom manipulator and collision/contact detection of contact between the log and ground and between the manipulator and log.[2] The dynamic models of machines become complicated when the manipulators are attached to a moving base. The moving base in the iterative Newton–Euler dynamic model can be described by employing a symbolic approach in the implementation.[3]

On the other hand, a conventional linear graph method can also be implemented for the models of

[1]Department of Mechanical Engineering, Lappeenranta University of Technology, Finland
[2]MeVEA Ltd, Finland

Corresponding author:
Mohamad Ezral Baharudin, Department of Mechanical Engineering, Lappeenranta University of Technology, Skinnarilankatu 34, Lappeenranta 53850, Finland.
Email: ezral.baharudin@lut.fi

the mechanical structure and hydraulic actuators of a harvester machine manipulator.[4] In the method, each single component is identified together with its constitutive equations. Linear graphs for each subsystem provide detailed variables and can be interconnected with other subsystems if they share variables. However, three-dimensional models produced using this approach result in less efficient dynamic modelling than models produced using a recursive multibody formulation.[5,6] To accelerate the computing time, it also possible to reduce the electro-hydraulic actuator model where the dynamics of fluid lines are eliminated and modelled as a purely resistive network.[7,8] The hydraulic system can also be modelled using a simple computing algorithm and self-tuning model reference adaptive control.[9]

Another almost similar study on developing the dynamics model and simulation of a heavy machine manipulator deals with cherry pickers where the system consists of a truck (base), two flexible arms and a passenger basket. The system is characterised by the two-mode vibrations of a zero vibration input shaper and an extra-insensitive input shaper that significantly reduces the motion-induced oscillation of the system.[10,11] The dynamic behaviour of the machine suspension can be modelled using an active, semi-active or passive suspension system solution, as explained in the study of Grott et al.[12]

In conclusion, many factors may affect the computing time in order for it to approach real-time application, such as the method of implemented dynamics, the integration time step, the numerical integration algorithm, the CPU type, the source code structure, the computing processing method and so forth[13] and the key to success in developing the harvester simulator is the simulation models of manipulators, vehicle motion due to unmade surface, cutting and loading the log.[14]

The primary objective of the work described in this article is to introduce a new and accurate simulation approach that closely models the performance of a complex machine system in a real environment and with representative feedback. This approach was implemented to develop a tree harvester simulator, and the modelling of dynamics of the harvester's mechanics, hydraulic actuators and contact phenomena are discussed in depth in the following sections. The mechanical system was modelled dynamically using a semi-recursive formulation and solved via parallel computing techniques. The equations of motion were developed recursively for an open loop (chain topology) rigid body system. The hydraulic actuators were modelled using lumped fluid theory, in which the hydraulic circuit volume is divided into discrete volumes, and pressures are assumed to be distributed equally. The penalty method was applied to define model collision and contact. The tires-ground contact is modelled using the LuGre friction model. The introduced simulation approach, applied to produce real-time simulation models for a tree harvester, is described in 'Numerical example'.

Mechanical model of tree harvester

Mechanical structure of the tree harvester is modelled using multibody system dynamics approach. The approach can be applied to dynamic analysis of systems consisting of bodies that are constrained to move relative to one another using kinematic connections. The system may have a tree or chain structure (open loop) or a structure with closed links of bodies (closed loop). In the chain structure system, the link between any two arbitrary bodies in the system is unique. As an example, a tree harvester truck can be simplified into a chain structure multibody system, as shown in Figure 1.

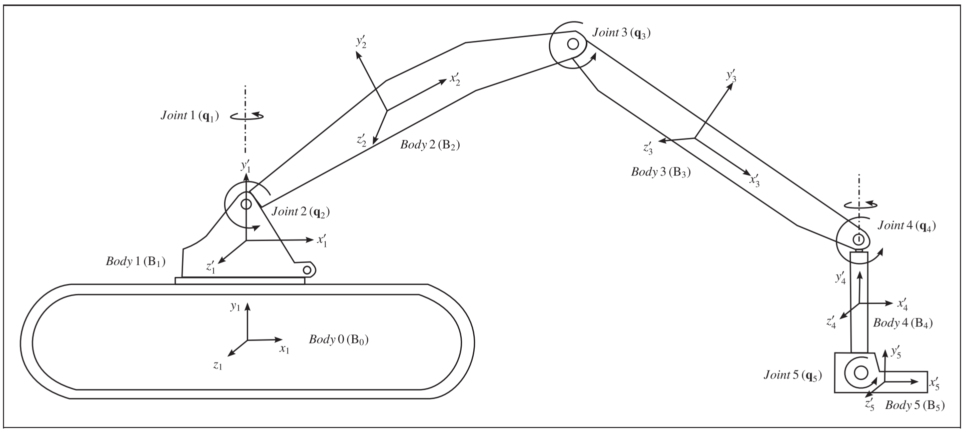


Figure 1. Tree harvester illustrated as multiple rigid bodies.

The tree harvesting mechanism, which excludes the vehicle body, comprises five bodies. Bodies 1, 2, 3, 4 and 5 are connected to the preceding bodies with revolute joints. The vehicle body B_0 is designated the base body. Body B_1 is attached to body B_0 by a revolute joint q_1. The fourth joint q_4 consists of two revolute joints – one vertical and one horizontal. Joints q_2, q_3 and q_5 are revolute joints. The mechanical system has six degrees of freedom. Because movement is possible in both the horizontal and vertical directions, the motion of the mechanism is three dimensional.

Recursive formalism

The recursive formulation for this study derives from the equations of motion for a constrained mechanical system. The kinematic properties such as position, velocity and acceleration are developed based on the relative coordinates between contiguous bodies connected by a joint.[15] Hooker and Margulies[16] proposed this formulation to the dynamic analysis of satellites and found the computational costs to increase linearly with the number of bodies. This algorithm has been used and extended by several researchers and has recently been generalised to improve its implementation and efficiency.[17–20] The dynamics equation in a recursive formulation is written in terms of the system's degrees of freedom and, typically, is written with a lower dimensionality than those in an augmented formulation.

As shown in Figure 1, bodies are connected with joints. Each body number can be represented as B_n. Preceding bodies are designated B_{n-1}. The arranged set of body numbers is called a 'body connection array'.[21] Joint numbers are named before body numbers. The revolute, prismatic, cylindrical, translational or spherical joints can consist of single or multiple axes.

Kinematics. Relative motion between neighbouring bodies and constraints are the two main aspects of recursive kinematics used to generate the total system matrices and solve the equations of motion for the multibody system. Figure 2 shows an elementary system of two bodies interconnected by a joint.

The orientation relationship of contiguous bodies is obtained by sequentially transforming from the body reference frame on body B_0 to the body reference frame on body B_n. The system of bodies is considered an open chain and point 0 is the global reference frame. The relative kinematics of position r_n for each body can be described with respect to the global reference as follows[22]

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{s}_{n-1} + \mathbf{d}_n \tag{1}$$

where $\mathbf{d}_n$ is the joint relative displacement vector from point $n-1'$ to n, and $\mathbf{s}_{n-1}$ is a vector from point $n-1$ to $n-1'$.

The orthogonal rotation matrix $\mathbf{A}_{n-1}$ is obtained from the preceding body B_{n-1}. The velocities can be determined from the time derivative of equation (1)

$$\mathbf{v}_n = \dot{\mathbf{r}}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} + \dot{\mathbf{d}}_n \tag{2}$$

$$\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} + \boldsymbol{\omega}_d \tag{3}$$

where matrix $\tilde{\boldsymbol{\omega}}_{n-1}$ is the skew-symmetric matrix of the cross-product $\tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} \equiv \boldsymbol{\omega}_{n-1} \times \mathbf{s}_{n-1}$. Vector $\boldsymbol{\omega}_d$ is the relative angular velocity of the joint. The acceleration equations can be obtained by differentiating equations (2) and (3) as follows

$$\ddot{\mathbf{r}}_n = \ddot{\mathbf{r}}_{n-1} + \dot{\tilde{\boldsymbol{\omega}}}_{n-1}\mathbf{s}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1}\tilde{\boldsymbol{\omega}}_{n-1}\mathbf{s}_{n-1} + \ddot{\mathbf{d}}_n \tag{4}$$

$$\dot{\boldsymbol{\omega}}_n = \dot{\boldsymbol{\omega}}_{n-1} + \dot{\boldsymbol{\omega}}_d \tag{5}$$

In equations (1) to (5), the terms $\mathbf{d}_\mathrm{n}$, $\mathbf{A}_d$ and $\boldsymbol{\omega}_d$ are functions of relative joint position, the joint rotation matrix and joint angular velocity, respectively. The functions of velocity and acceleration for each common type of joint (prismatic, spherical, etc.) can be obtained by deriving expressions as suggested by Avello et al.[23]

Equation of motion. The equations of motion can be developed from the principle of virtual power, as proposed by Avello et al.[23] and Haug.[24] In Figure 2, if position vector r_n is shifted to the centre of gravity of the body becoming $\mathbf{u}_n$, $\boldsymbol{\omega}_n$ is the velocity of body n and N_b is the number of contiguous bodies. The equation of motion can be written in matrix form as follows

$$\sum_{n=1}^{N_b}\left\{\delta\dot{\mathbf{u}}_n^{\mathbf{T}}\delta\boldsymbol{\omega}_n^{\mathbf{T}}\right\}\left(\begin{bmatrix}\mathbf{m}_n\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_n\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{u}}_n\\ \dot{\boldsymbol{\omega}}_n\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \tilde{\boldsymbol{\omega}}_n\mathbf{J}_n\boldsymbol{\omega}_n\end{bmatrix} - \begin{bmatrix}\mathbf{F}_n\\ \mathbf{T}_n\end{bmatrix}\right) = 0 \tag{6}$$

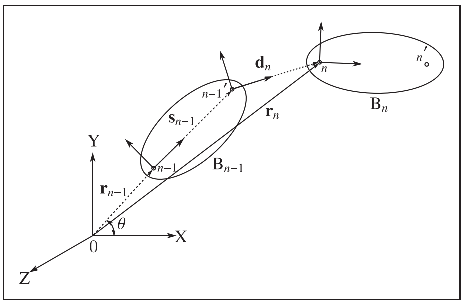


Figure 2. Relationship of contiguous bodies.

where $\delta\dot{\mathbf{u}}_n$ and $\delta\boldsymbol{\omega}_n$ are virtual translational and angular velocities, matrix $\mathbf{J}_n$ is inertia (obtained from $\mathbf{A}_n\mathbf{J}_n\mathbf{A}_n^{\mathbf{T}}$), $\mathbf{m}_n$ is body mass and $\mathbf{F}_n$ and $\mathbf{T}_n$ are external forces and torques. The equations of motion can be simplified as

$$\delta\dot{\mathbf{q}}^{\mathbf{T}}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} - \mathbf{Q}) = 0 \tag{7}$$

From equation (7), the dependent virtual velocities $\delta\dot{\mathbf{q}}$ should be eliminated to obtain the final equation with a small dimension. Therefore, the vector of independent virtual velocities $\delta\dot{\mathbf{z}}$ with N_f number of degrees of freedom is introduced. As mentioned previously, $\dot{\mathbf{q}}$ has a dimension of $6N_b$, which covers the translational and angular velocities of each body. However, the angular position cannot be obtained by integrating $\dot{\mathbf{q}}$, which necessitates the introduction of the new variable $\mathbf{p}_q$ to represent the dependent position. The new variable can be defined by separately defining the translational and orientation as suggested in the Cartesian coordinate approach[25] or other suitable methods. In this case, $\dot{\mathbf{q}}$ and $\dot{\mathbf{p}}_q$ can be represented as follows

$$\dot{\mathbf{p}}_q = \mathbf{E}_q\dot{\mathbf{q}} \tag{8}$$

$$\dot{\mathbf{q}} = \mathbf{F}_q\dot{\mathbf{p}}_q \tag{9}$$

In the preceding equations, $\mathbf{E}_q$ is the velocity transformation matrix and $\mathbf{F}_q$ is the inverse velocity transformation matrix. Similar to $\dot{\mathbf{q}}$, $\dot{\mathbf{z}}$ cannot be integrated to obtain a position, which necessitates introducing the position vector $\mathbf{p}_z$ and the velocity transformation matrix $\mathbf{E}_z$ with the size $(N_f \times N_f)$

$$\dot{\mathbf{p}}_z = \mathbf{E}_z\dot{\mathbf{z}} \tag{10}$$

In this particular equation, $\mathbf{p}_q$ can be represented as a function of $\mathbf{p}_z$ and time as follows

$$\mathbf{p}_q = \mathbf{p}_q(\mathbf{p}_z, t) \tag{11}$$

By taking the time derivative of equation (11), the final $\dot{\mathbf{q}}$ can be obtained as

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} + \mathbf{b} \tag{12}$$

where

$$\mathbf{R} = \mathbf{F}_q\frac{\partial\mathbf{p}_q}{\partial\mathbf{p}_z}\mathbf{E}_z \text{ and } \mathbf{b} = \mathbf{F}_q\frac{\partial\mathbf{p}_q}{\partial t}$$

The acceleration equation can be formulated by taking the time derivative of equation (12)

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}} \tag{13}$$

Substituting equation (13) into the original equation (7) leads to these expressions

$$\delta\dot{\mathbf{z}}^{\mathbf{T}}(\mathbf{R}^{\mathbf{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} + \mathbf{R}^{\mathbf{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{R}^{\mathbf{T}}\mathbf{M}\dot{\mathbf{b}} + \mathbf{R}^{\mathbf{T}}\mathbf{C} - \mathbf{R}^{\mathbf{T}}\mathbf{Q}) = 0 \tag{14}$$

Equation (14) can be reconstructed in simpler form as follows

$$\begin{aligned}\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} &= -\mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} - \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{b}} - \mathbf{R}^{\mathrm{T}}\mathbf{C} + \mathbf{R}^{\mathrm{T}}\mathbf{Q}\\ &= \mathbf{R}^{\mathrm{T}}(\mathbf{Q} - \mathbf{C}) - \mathbf{R}^{\mathrm{T}}\mathbf{M}(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}})\end{aligned} \tag{15}$$

Computing matrix R. Here, it is clearly shown that velocity transformation matrix **R** plays an important role. Consequently, it is important to pay attention to solving the matrix **R** that affects each body. For example, referring to the simple analogy of Figure 1 and assuming that all joints have one degree of freedom, the total size of matrix **R** is (30 × 5), where the columns of matrix **R** correspond to each joint in the system. If body B_3 has been chosen to identify matrix R_3, the columns corresponding to joints 4 and 5 have all zero elements and only the 1, 2 and 3 joints need be computed. Therefore, matrix $\mathbf{R}_3$ is a (6 × 3) matrix $\left[\mathbf{R}_3^1\mathbf{R}_3^2\mathbf{R}_3^3\right]$. The superscripts represent the joint number. Each submatrix is a (6 × 1) vector and the first three components of the vector are obtained from equation (2). The last three components come from equation (3). From here, it can be generalized that $\mathbf{R}_n$ can be formed into $\left[\mathbf{R}_n^1\mathbf{R}_n^2\ldots\mathbf{R}_n^{N_q}\right]$ with a size of $(6 \times N_d)$, where N_d and N_q are the number of degrees of freedom and number of joints from body n to the vehicle body, respectively. The size of the sub-matrix $\mathbf{R}_n^j$ is $(6 \times N_j)$, where N_j is the number of degrees of freedom at joint j.

Numerical algorithm for open loop system. The system of differential algebraic equations has been transformed into ordinary differential equations by integrating variables $\ddot{\mathbf{z}}$ and $\dot{\mathbf{p}}_z$. In general, the algorithm is derived for generating the real-time simulation for an open loop system. The topology of the system should be defined first by labelling all of the bodies and joints. The algorithm to recursively compute the dynamics of the open loop system is started by assigning the initial values for positions $\mathbf{p}_z$ and velocity $\dot{\mathbf{z}}$ at $t = 0$. Then, by solving the position problem, $\mathbf{r}_n$, $\mathbf{q}_n$ and $\mathbf{A}_n$ can be obtained. Subsequently, the velocities $\dot{\mathbf{r}}_n$, ω_n and $\left(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}}\right)$ are computed as a velocity-dependent acceleration. With the help of a parallel computing loop from $n = 1{:}N_b$:

1. Compute $\mathbf{M}_n$, $\mathbf{C}_n$, $\mathbf{Q}_n$ and finally $\mathbf{R}_n$
2. Compute $\mathbf{R}_n^{\mathbf{T}}(\mathbf{Q}_n - \mathbf{C}_n)$ and $\mathbf{R}_n^{\mathbf{T}}\mathbf{M}_n^{\mathbf{T}}$
3. Compute $\left(\mathbf{R}_n^{\mathbf{T}}\mathbf{M}_n\right)\mathbf{R}_n$ and $\left(\mathbf{R}_n^{\mathbf{T}}\mathbf{M}_n\right)\left(\dot{\mathbf{R}}\dot{\mathbf{z}} + \dot{\mathbf{b}}\right)$.

By using equation (25), the acceleration of $\ddot{\mathbf{z}}$ can be obtained. Compute $\dot{\mathbf{p}}_z = \mathbf{E}_z\dot{\mathbf{z}}$ and by integrating $\left(\ddot{\mathbf{z}}, \dot{\mathbf{p}}_z\right)$, $\left(\dot{\mathbf{z}}, \mathbf{p}_z\right)_{t+\delta t}$ can be obtained. In this algorithm, the parallel computing loop requires the most computing costs and may cause problems if not handled accurately.

Modelling of actuators

In this study, hydraulic actuators, the assumed drivers for the tree harvester systems, are modelled using lumped fluid theory, which divides a hydraulic circuit into discrete volumes with hydraulic pressure distributed equally. Valves are modelled using a semi-empirical approach that makes use of flow parameters that can be obtained, in many cases, directly from manufacturer catalogues. Usually, a hydraulic system has a high nominal frequency response and, therefore, the calculation time step must be short to produce reliable results. In this approach, pressure wave propagation in pipes and hoses is assumed to be negligible.[26] The hydraulic pressure in each hydraulic volume i can be described as follows

$$\dot{p}_l = \frac{B_{ei}}{V_i}\sum_{j=1}^{n_c} Q_{ij} \tag{16}$$

where B_{ei} is the effective bulk modulus that defines the flexibility of the hydraulics, Q_{ij} is the outgoing or incoming flow rate and n_c is the total number of hydraulic components, all related to volume i, V_i. The effective bulk modulus can be calculated as

$$B_{ei} = \frac{1}{\frac{1}{B_{oil}} + \sum_{j=1}^{n_c} \frac{V_j}{V_i B_j} + \sum_{k=1}^{n_h} Q_{ij} \frac{V_k}{V_i B_k}} \tag{17}$$

In equation (17), n_h is the total number of pipes and hoses related to volume i. The bulk modulus of oil, B_{oil}, accounts for the amount of non-dissolving air in the oil. It is a function of pressure and the maximum value is typically $B_{oilmax} = 1.6 \times 10^9$ Pa. The bulk moduli B_j and B_k of components j and k are also dependent on the component type.

The magnitude of volume V_i is solved based on the combined volume of pipes, hoses and other hydraulic components related to the volume as follows

$$V_i = \sum_{j=1}^{n_c} V_j + \sum_{k=1}^{n_h} V_k \tag{18}$$

where the volume V_j of a single component depends on the types of the component. The bulk modulus B_j of component j is also dependent on the component type.

Modelling of valves

For modelling purposes, a valve is assumed to consist of several adjustable restrictor valves that can each be modelled separately.[27] Figure 3 shows an example of 4/3 directional valve where flow rate at each port (P_A, P_B, P_T and P_P) can be calculated. With small pressure differences (< 1 bar), the flow over the restrictor is thought to be laminar and with larger differences, it is thought to be turbulent. When using the semi-empiric modelling method, the flow over the restrictor can be written as follows

$$Q = C_v U \sqrt{dp} \tag{19}$$

where C_v is the flow rate constant defining the size of the valve, U is a variable that defines the spool or poppet position and dp is the pressure difference between pressure ports. For a number of valve types, the variable U can be defined using a first-order differential equation

$$\dot{U} = \frac{U_{ref} - U}{\tau} \tag{20}$$

where U_{ref} is the spool reference position and τ is the time constant describing the dynamics of the valve spool.

Modelling of hydraulic cylinders

A hydraulic cylinder can be modelled using the dimensions of the cylinder and the pressure obtained from equation (16). The volumes of hydraulic cylinder chambers are solved as a function of cylinder stroke as follows

$$\begin{aligned} V_{jA} &= xA_A \\ V_{jB} &= (l - x)A_B \end{aligned} \tag{21}$$

where A_A is the area of the cylinder piston side and A_B is the area on the cylinder piston rod side, x is the stroke of the cylinder and l is the cylinder maximum stroke. The motion of the hydraulic cylinder produces a flow rate Q to the hydraulic volume i as follows

$$\begin{aligned} Q_{jA} &= -\dot{x}A_A \\ Q_{jB} &= \dot{x}A_B \end{aligned} \tag{22}$$

The stroke velocity of the cylinder is $\dot{x}$, as depicted in Figure 4.

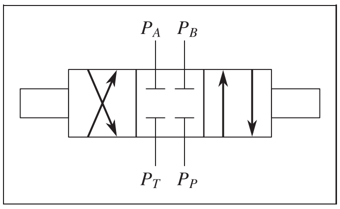


Figure 3. Direction valve.

The force produced by the hydraulic cylinder F_s can be defined as

$$F_s = p_1 A_1 - p_2 A_2 - F_\mu \tag{23}$$

where F_μ is the total friction force of the cylinder and p_1 and p_2 are pressures acting in the cylinder chambers. Friction force is a function of pressures, cylinder efficiency η and velocity. The friction force can be described in a simple case as follows

$$F_\mu = (p_1 A_1 - p_2 A_2)(1 - \eta) f(\dot{x}) \tag{24}$$

The velocity-dependent friction coefficient, $f(\dot{x})$, can be described using a spline curve, as shown in Figure 5.

In this particular model, the maximum speed of the hydraulic cylinder is 1 ms^{-1}, as shown in Figure 5. Friction is a physical process of shear between two sliding surfaces as a function of velocity. In the friction model used, the coefficient decreases at a low velocity and increases proportionally when the velocity increases after a specific velocity.[28] The minimum velocity-dependent friction coefficient is 0.2 at 0.1 ms^{-1}. Hydraulic cylinder end damping is modelled using impact force as follows

$$\begin{aligned} F_{Imin} &= k(x - x_{min}) + c\dot{x}\ \mathrm{STEP}\ (\dot{x},\ -1e-3, 0, 1, 0), \\ &\quad x < x_{min} \\ F_{Imax} &= k(x - x_{max}) + c\dot{x}\ \mathrm{STEP}\ (\dot{x}, 0, 1e-3, 0, 1), \\ &\quad x > x_{max} \end{aligned} \tag{25}$$

where k is the end damping spring constant and c is the damping coefficient. The STEP function that uses a cubic polynomial to fit between two states is used in the formulation code in order to avoid the instantaneously contact force.[29]

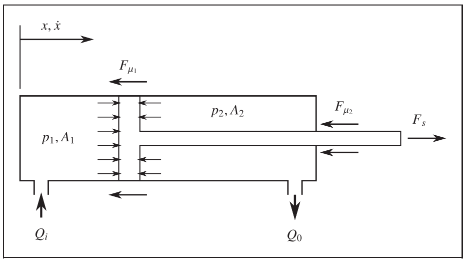


Figure 4. Hydraulic cylinder transforms hydraulic pressure into a mechanical force.

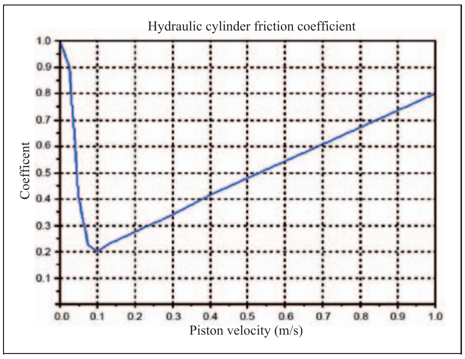


Figure 5. The velocity-dependent friction co-efficient.

Collision and contact modelling

The contact model is important and must be considered when simulating more than one body. Its function is to prohibit the interpenetration of bodies. The contact algorithm determines when and where contact occurs and calculates the contacting elements.

In practice, there are two steps in contact modelling: contact detection and collision response.[30] Contact detection is the first and most important step. It models the moving bodies and determines when and where a potential collision will occur. The accuracy of the results depends on the geometry of the contacting bodies. Contact response prevents penetration between bodies by analysing the contact force of colliding bodies based on their geometric properties, relative velocities and material makeup. However, interbody penetration between rapidly moving bodies may occur between times steps if excessive time steps are being used.[31]

There are several ways to define contact in multibody systems, including the penalty technique, analytical method and impulse method.[32–34] In this study, contacts between bodies are defined using the penalty method, which allows small penetrations between bodies, and a temporal spring damper is added between the collision points.[35–37] Since it permits interpenetration,[38] the penalty method is also known as the surface compliance or soft contact method. The general algorithm for contact/collision models is illustrated by the state transition diagram in Figure 6.

In Figure 6, d and $\dot{d}$ are the distance and velocity between two objects, while F_N and F_T are normal and tangent forces. The idea of this algorithm is to always compare the distance between two bodies and a collision occurs when the distance d is equal to or less than 0. Then, the collision response model will be implemented. The contact force is applied to the model as an external force.

In this study, the object-oriented bounding box (OOBB) model is used for collision detection and the penalty method is applied as a collision response between the harvester head and log. Contact between tires and the ground is modelled using the LuGre model as a friction model.

Collision detection

The collision between bodies with different geometries is detected using the bounding volume (BV) approach. This method uses detection trees to determine whether or not proximate bodies are intersecting. The BV approach uses simple bounding volumes such as spheres and boxes to encapsulate the more complex body geometries. The simplicity and tightness of the bounding box affect the efficiency of collision detection. There are a few types of bounding methods available for collision detection, such as bounding spheres,[39] an axis-aligned bounding box,[40] a discrete-orientation polytope[41] and an oriented bounding box.[42]

In this study, collision detection is accomplished by employing the OOBB approach. This approach offers the minimum rectangular solid embodying the object and along the direction of axis. The tightness of OOBB can significantly reduce the number of bounding boxes involved in intersection detection. If the object has variety of shapes, the OOBB can be divided into small boxes that are connected to each other, that is, an OOBB tree. Each box can be treated separately for collision detection. However, the higher the number of OOBBs in the hierarchy, the greater the computing costs. The separating axis theorem is used to test whether the two OOBBs on an axis either intersect or not. For a three-dimensional case, this theorem offers 15 potential separating axes, which need to be examined, and if overlapping occurs on every single separating axis, the boxes intersect. These are the three local axes of the first OOBB, the three local axes of the second and the nine cross-products of an axis of the first OOBB with an axis of the second. If the boxes do not intersect, the given axis is called as a separating axis. Figure 7 shows an OOBB collision detection approach in a two-dimensional case between harvester head and log.

In Figure 7, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{B}_1$ and $\mathbf{B}_2$ are normalised axes of boxes A and B while a_1, a_2, b_1 and b_2 are the radii of boxes A and B. **L** is a normalised direction and T is the distance from the centres of A and B. The formulations for p_A and p_B are

$$\begin{aligned} p_A &= a_1\mathbf{A}_1\mathbf{L}_i + a_2\mathbf{A}_2\mathbf{L}_i \\ p_B &= b_1\mathbf{B}_1\mathbf{L}_i + b_2\mathbf{B}_2\mathbf{L}_i \end{aligned} \tag{26}$$

Boxes A and B do not overlap in a two-dimensional case if $\mathbf{T.L} > p_A + p_B$. For a three-dimensional case, all 15 separating axes must be tested and the two OBBs are separated if

$$\mathbf{T} \cdot \mathbf{L}_i > p_{Ai} + p_{Bi} \quad i = 1:15 \tag{27}$$

Under certain conditions, collision detection needs special treatment, for example, when dealing with particles. Even though this collision detection method will affect computing costs, especially when a large number of particles are involved, it will produce good results for extended simulation. In general, particle-based collision detection can be divided into several steps and involves different approaches such as the neighbour search method or the force model.[43]

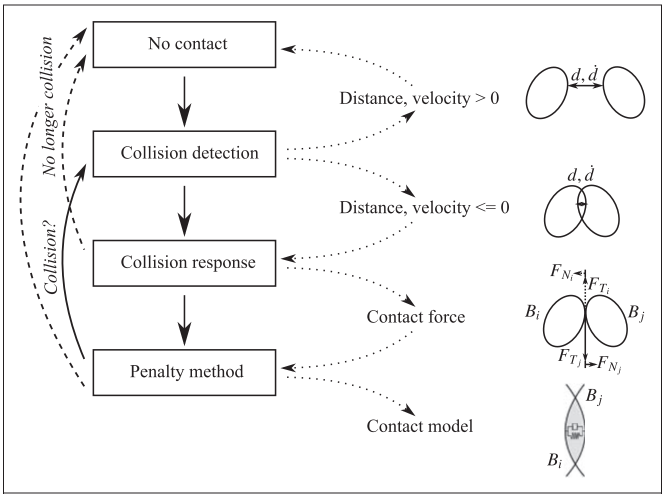


Figure 6. State transition diagram track of contact phase between two bodies.

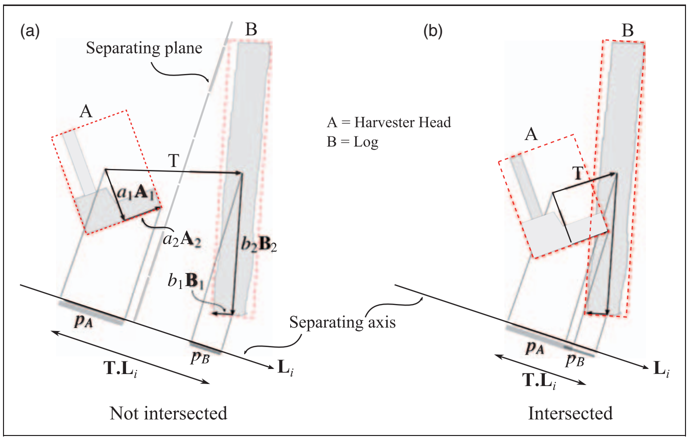


Figure 7. Object Oriented Bounding Box (OOBB): (a) not intersected and (b) intersected.

Collision response

In practice, it is difficult to avoid body penetration even when using smaller time steps in the collision prediction algorithm. Using smaller time steps reduces penetration occurrences but increases computing time. Hence, there is a limit to how small the time steps can be made. This limitation may result in inaccurate simulation results. Therefore, applying the penalty method could produce an acceptable result in the real-time environment. The penalty method can be categorised into two types: the normal force methods and the geometry-based model. The normal force method deals with the single force model and will account for the entire collision. The geometry-based model can be divided into single collision point models, multiple collision point models and volumetric models.[44–48] In this particular study, the geometry-based model (single point) was chosen.

The kinematics of collision points can be described using a global reference frame. Figure 8 gives an example of collision direction before and during a collision.

In Figure 8, F_t and F_n are the tangential normal force at the collision point. Distance d can be written from the basic global reference frame approach as follows

$$\begin{aligned}\mathbf{r}_{P_A} &= \mathbf{r}_A + \mathbf{A}_A\bar{\mathbf{u}}_A \\ \mathbf{r}_{P_B} &= \mathbf{r}_B + \mathbf{A}_B\bar{\mathbf{u}}_B\end{aligned} \tag{28}$$

$$\mathbf{d} = \mathbf{r}_{P_B} - \mathbf{r}_{P_A} = \mathbf{r}_B + \mathbf{A}_B\bar{\mathbf{u}}_B - \mathbf{r}_A - \mathbf{A}_A\bar{\mathbf{u}}_A \tag{29}$$

where $\mathbf{A}_A$ and $\mathbf{A}_B$ are a rotation matrix of bodies A and B in three dimensions and $\bar{\mathbf{u}} = [x\ y\ z]^T$ is a relative position vector of bodies A and B. Knowing that the relative velocity is the time derivative of d, the relative velocity is

$$\begin{aligned}\mathbf{v}_{AB} &= \dot{\mathbf{r}}_B + \tilde{\boldsymbol{\omega}}_B\bar{\mathbf{u}}_B + \mathrm{A}_B\dot{\bar{\mathbf{u}}}_B - \dot{\mathbf{r}}_A - \tilde{\boldsymbol{\omega}}_A\bar{\mathbf{u}}_A - \mathrm{A}_A\dot{\bar{\mathbf{u}}}_A \\ &= \dot{\mathbf{r}}_B - \tilde{\bar{\mathbf{u}}}_B\boldsymbol{\omega}_B + \mathbf{H}_B\dot{\mathbf{q}}_B - \dot{\mathbf{r}}_A + \tilde{\bar{\mathbf{u}}}_A\boldsymbol{\omega}_A - \mathbf{H}_A\dot{\mathbf{q}}_A\end{aligned} \tag{30}$$

As mentioned earlier, a spring and damper are added in the penalty method in order to define the contact forces when a small penetration is allowed between contacting bodies. Basically, the contact force at the contact point can be formulated as

$$\mathbf{F} = -K\mathbf{x} - C(\mathbf{n} \cdot \mathbf{v}_{AB}) \tag{31}$$

where $\mathbf{n}$ is the contact normal, $\mathbf{x}$ is the penetration depth, K is the coefficient of elasticity and C is the damping factor. The dot product of relative velocity $\mathbf{v}_{AB}$ and contact normal $\mathbf{n}$ is the component of the relative velocity in the direction of the collision normal. The choice of value for the two coefficients K and C is vital and may result in a range of desired collision response types.

Tire modelling

The harvester truck will manoeuvre on the unmade surfaces/off road and each movement will affect the dynamic behaviour of the system. There are different approaches to model the contact between the tires and ground and the most realistic model should consider the contact patch interaction between the tires and soil using finite element method and multibody dynamics.[49,50] However, simulating the detail dimensions of the contact patch between the tires and ground surface/soil will increase the computing

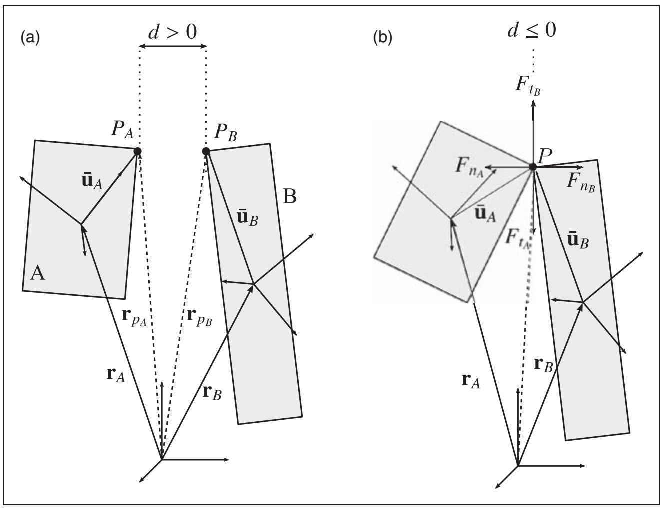


Figure 8. Collision between bodies A and B.

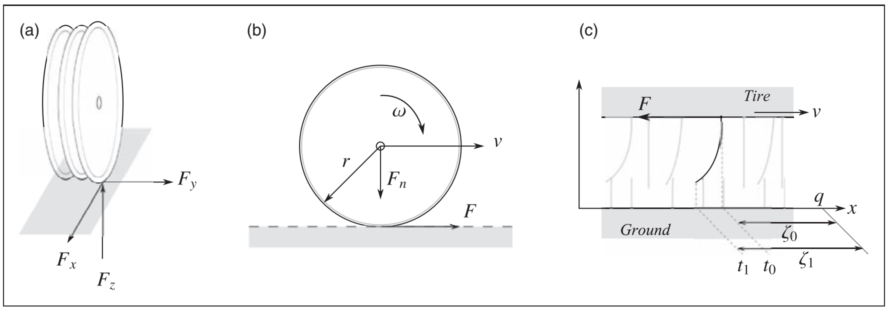


Figure 9. LuGre tire model.

costs, which is not desirable for the real-time simulation. Therefore, in order to meet the real-time criterion, the tires and ground are assumed to be simple rigid bodies and the profile of the ground can be constructed depending on the required environment (off-road profile).

For the implementation of this particular tree harvester simulator, the tire of the truck is presumed as a series of discs, as shown in Figure 9(a), whereas the typical forces involved are longitudinal force F_x, lateral force F_y and vertical force F_z. In this simulation, the contact between tire and ground is assumed as contact between two rigid bodies. Figure 9(b) shows the friction model with punctual contact, whereas F_n is the normal force, F is the friction force and v is the relative velocity between the bodies.

The lumped LuGre model is used to model the contact between tire and ground. The LuGre friction model is an enhancement of the Dahl model with the Stribeck effect added. This model captures many important aspects of friction, such a stiction, the Stribeck effect, stick slip, zero slip displacement and hysteresis.[51–54] The contact between tire and ground is modelled as two rigid bodies that contact through an elastic bristle at the microscopic level, as shown in Figure 9(c). When contact occurs, the tangential force will affect the bristles by deflecting like springs and create the friction force.[55,56] Whena large force is imposed during the contact, some of the bristles could deflect excessively and start to slip. When the tire rotates, the bristle will deflect with respect to time and the deflection of the bristles denoted by z is derived by defining the normalised longitudinal as

$$z = \frac{\zeta_1 - \zeta_0}{\zeta_0} \tag{32}$$

where ζ_1 is the distance from the deformed bristle point to point q while ζ_0 is the distance from an undeformed bristle point to point q. The average deflection of bristles is the time derivative of z as

$$\frac{dz}{dt} = v - \frac{\sigma_0 |v|}{g(v)} z \tag{33}$$

where v is the relative velocity between two sliding surfaces. This shows that the deflection is relative to the integral of the relative velocity. The function g (friction) must always be positive and depends on material properties, lubricants, temperature and other factors. The friction force generated from the bending bristles is

$$F = \left(\sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 v \right) F_n \tag{34}$$

where σ_0 is the longitudinal lumped stiffness, σ_1 is the longitudinal lumped damping coefficient and σ_2 is viscous relative damping. The Stribeck effect occurs

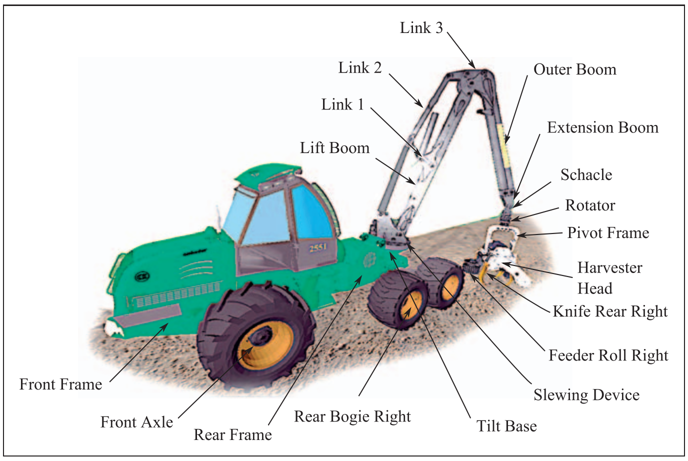


Figure 10. Tree harvester connecting components.

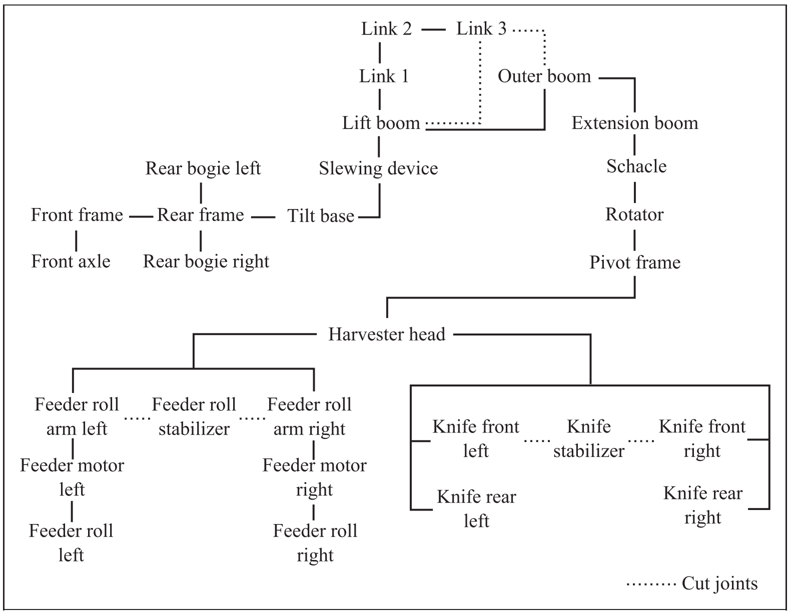


Figure 11. Topology of tree harvester model.

when the typical friction $g(v)$ decreases with increasing velocity v for low velocities. Therefore, function $g(v)$, which describes the Stribeck sliding friction force effect is

$$g(v) = F_c + (F_s - F_c)e^{-\left|\frac{v}{v_s}\right|^{1/2}} \tag{35}$$

where F_c is Coulomb friction, F_s is normalized static friction and v_s is the Stribeck relative velocity.

Numerical example

The following numerical example is based on a typical tree harvester, as shown in Figure 10. The simulation model of the harvester consists of a number of bodies and is modelled using a recursive method. Figure 11 shows the simplified topology of the harvester used as a reference when implementing the recursive method. The models include several force components such as booms, hydraulics and log cutters (harvester head).

Table 1. Summary of model.

Name	Number
Number of bodies	25
Number of joints	29
Number of joint coordinates	31
Number of closed loop constraint equations	7
Degrees of freedom	23
State variables	177

The mechanics of a tree harvester can be considered as a set of rigid bodies with mechanical joints limiting the relative motion of the coupled bodies. Configuration changes with time based on the forces and motions applied to the harvester's components. Kinematic models consider system body motions without taking into account the forces that

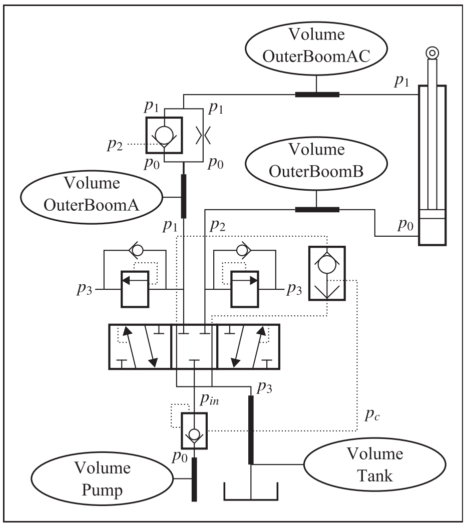


Figure 12. Simplified description of outer boom hydraulic system.

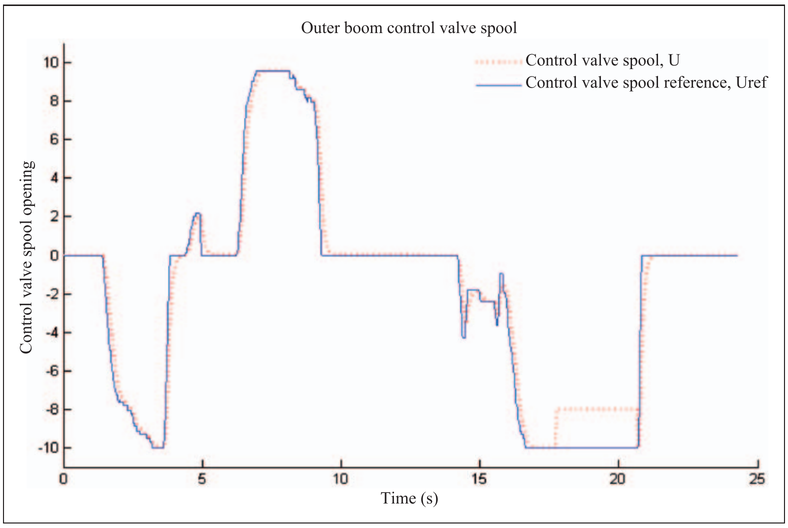


Figure 13. Control valve signal and reference control signal.

produce the motion. A kinematic model shows how the bodies move within the system. It is solved using linear and non-linear systems of equations. However, a dynamic model must be used to reveal the movement of the bodies and the realistic responses of the system to applied and resultant forces. To develop the kinematic and dynamic model simulations, the structure of the machine needs to be identified early so that all simulation procedures are based on a common structure.

Figure 11 is the topology of the tree harvester model, which shows the structures are connected to

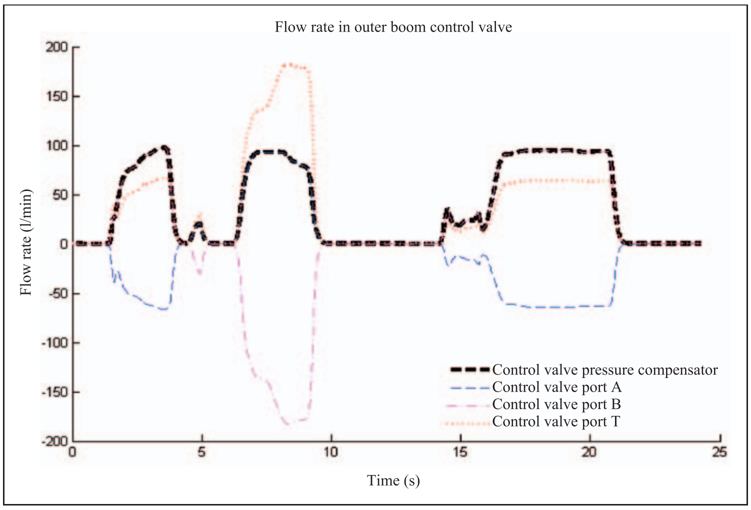


Figure 14. Flow rates through the valve during work cycle.

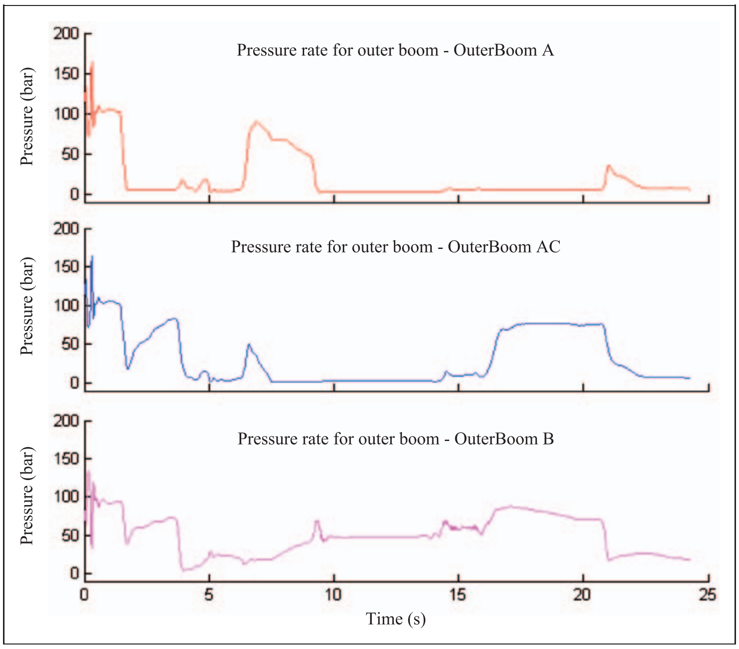


Figure 15. Pressure rates at the inlet (A and AC) and outlet (B) of the cylinder.

each other through joints from the bogie to the harvester head. There are two major types of joints in this structure: the revolute joint and the cylindrical joint. This structure consists of open loop and closed loop links. For the closed loop link, the cut joint is introduced and the penalty approach must be added as a constraint equation. For this particular tree harvester topology, the model structure can be summarised as in Table 1.

Since the model under investigation is used in training simulators, the hydraulic model example for this study is a simplified version of the actual hydraulics for the outer boom, as shown in Figure 12.

A hydraulic circuit consists of the hydraulic cylinder(s), the pressure-compensated proportional directional valve(s), the pressure relief valve(s) and the pump(s). Figures 13 and 14 give examples of a simple work cycle. Figure 13 shows the valve control signal and spool opening. Figure 14 shows the flow rates through the outer boom circuit valve. The pressure rates of the cylinder volumes are shown in Figure 15.

The collision simulation between the harvester head and ground is shown in Figure 16, where the harvester head is pressed against the ground and then the boom swing motion is added. The initial

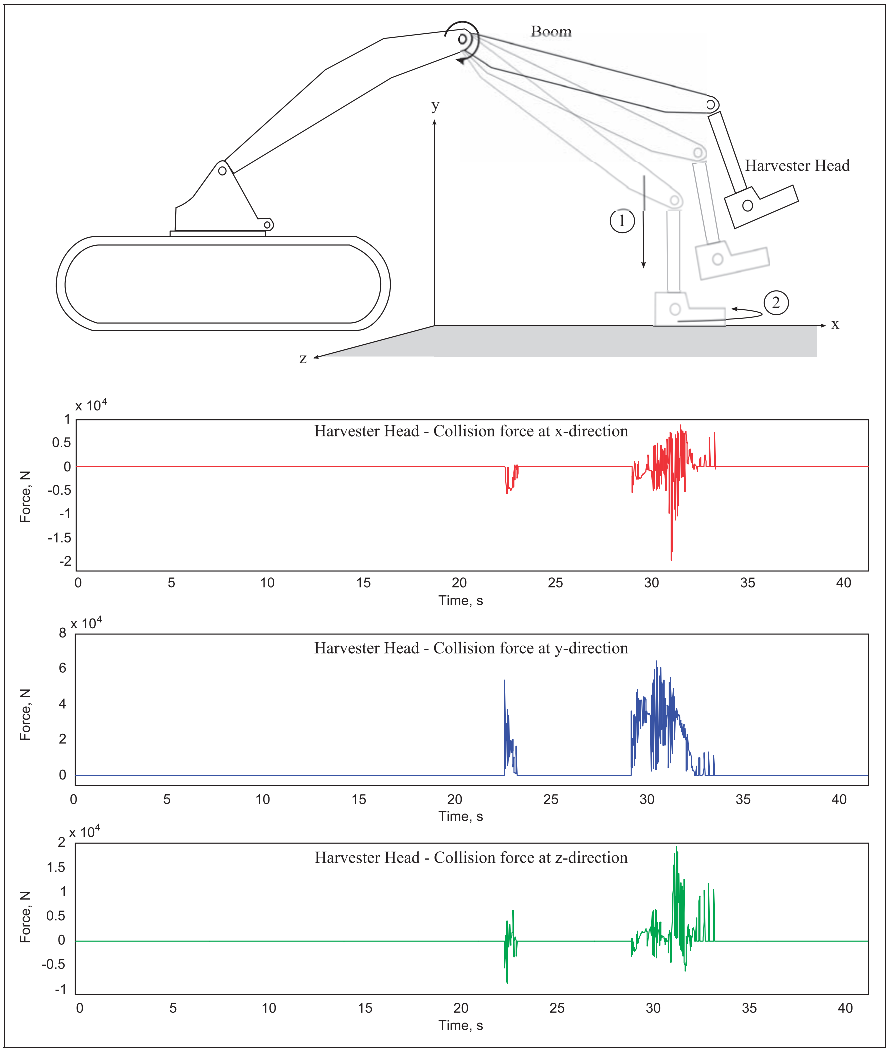


Figure 16. Collision force between harvester head and ground.

condition is that the boom points to the x-direction. It is lowered until the harvester head collides with the ground. The boom is swung to the left and the pressure towards the ground is constant.

The simulation was carried out using the MeVEA real-time simulation environment, which offers the possibility of offline simulation for a more detailed model or real-time simulation and visualisation for simplified models. The environment is compatible with the MeVEA Full Mission Solution, which offers a motion platform and visualisation environment combined with a user interface that includes a seat for the operator, joysticks, and pedal and so forth configured specifically for the application.[57]

Conclusions

In real-time simulation, a machine must be considered as a coupled system that consists of mechanical components and actuators. This study introduced a general simulation approach that can be applied to the real-time simulation of a hydraulically driven harvester. The introduced approach was based on the recursive method coupled with lumped fluid theory for the modelling of hydraulic circuits and the penalty method for contact modelling. The introduced simulation approach was applied to create real-time simulation models of a tree harvester. The simulation model of the harvester was installed and tested in a real-time simulation environment consisting of a visualisation display, a motion platform and an I/O interface. The real-time simulation environment merged the operator virtually with the simulated tree harvester.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Menghal PM and Jaya Laxmi A. Real time simulation: a novel approach in engineering education. In: *3rd international conference on electronics computer technology (ICECT)*, Kanyakumari, India, 8–10 April 2011, vol. 1, pp.215–219.
2. Freedman P, MacKenzie P and Lapointe JF. A computer-based training environment for forestry telemanipulation. In: *Proceedings of the international conference on intelligent robots and systems (IROS)*, Grenoble, France, 7–11 September 1997, vol. 3, pp.1826–1831.
3. Papadopoulos E and Sarkar S. The dynamics of an articulated forestry machine and its applications. In: *Proceedings of the IEEE international conference on robotics and automation*, Albuquerque, New Mexico, 20–25 April 1997, vol. 1, 323–328.
4. Papadopoulos E, Frenette R, Mu B, et al. On the modeling and control of an experimental harvester machine manipulator. In: *Proceedings of the international conference on intelligent robots and systems (IROS)*, Grenoble, France, 7–11 September 1997, vol. 1, pp.323–328.
5. Fisette P, Bruls O and Swevers J. Multiphysics modeling of mechatronic multibody systems. In: *Proceedings of the international conference on noise and vibration engineering (ISMA2006)*, Leuven, Belgium, 18–20 September 2006, pp.41–67.
6. Samin JC, Bruls O, Collard JF, et al. Multiphysics modeling and optimization of mechatronic multibody systems. *Multibod Syst Dynam* 2007; 18: 345–373.
7. Gonthier Y and Papadopoulos E. On the development of a real-time simulator for an electro-hydraulic forestry machine. In: *Proceedings of the IEEE international conference on robotics and automation*, Leuven, Belgium, 16–20 May 1998, vol. 1, pp.127–132.
8. Papadopoulos E, Mu B and Frenette R. On modeling, identification and control of a heavy-duty electrohydraulic harvester manipulator. *IEEE/ASME Trans Mechatron* 2003; 8(2): 178–187.
9. Chiang MH and Huang CC. Experimental implementation of complex path tracking control for large robotic hydraulic excavator. *Int J Adv Manufac Technol* 2004; 23: 126–132.
10. Jia H, Li W and Singhoose W. Dynamic modeling and vibration analysis of cherry pickers with flexible arm. *J Vib Shock* 2010; 29(12): 136–140.
11. Hongxia J, Wanli L and Singhose W. Using two-mode input shaping to suppress the residual vibration of cherry pickers. In: *3rd international conference on measuring technology and mechatronics automation*, Shanghai, China, 6–7 January 2011, vol. 3, pp.1091–1094.
12. Grott M, Biral F, Oboe R, et al. Semi-active suspension systems for heavy-duty vehicles: multibody model development, identification and control evaluation. In: *ASME international mechanical engineering congress and exposition, vol. 10: mechanical systems and control, Part A and B*, Lake Buena Vista, FL, USA, 13–19 November 2009, pp.13–19.
13. Sarkar S. *Dynamic modeling of an articulated forestry machine for simulation and control*. MS Thesis, McGill University, Montreal, Canada, 1996.
14. Freund E, Krämer M and Rosmann J. Towards realistic forest machine simulators. In: *AIAA modeling and simulation technologies conference and exhibit*, Denver, CO, 14–17 August 2000, paper no. AIAA-2000-4095, p.1.
15. Bae DS and Haug EJ. A recursive formulation for constrained mechanical system dynamics: part 1. Open loop system. *Mech Struct Mach* 1987; 15(3): 359–382.
16. Hooker W and Margulies G. The dynamical attitude equation for an n-body satellite. *J Astronaut Sci* 1965; 12: 123–128.
17. Featherstone R. The calculation of robot dynamics using articulated-body inertias. *Int J Robot Res* 1983; 2: 13–30.
18. Bae DS and Haug EJ. A recursive formulation for constrained mechanical system dynamic: part II. Closed loop system. *Mech Struct Mach* 1988; 15(4): 481–506.
19. Kim SS and Haug EJ. A recursive formulation for flexible multibody dynamics. Part I: open loop system. *Comput Methods Appl Mech Eng* 1988; 71: 481–506.
20. Bae DS, Han JM and Yoo HH. A generalized recursive formulation for constrained mechanical system dynamic. *Mech Struct Mach* 1999; 27(3): 293–315.

21. Slaats PMA. Recursive formulations in multibody dynamics. Report, computational mechanics, Technische Universiteit Eindhoven, December 1991.
22. Rahnejat H. *Multi-body dynamics: vehicles, machines and mechanism*. London: Professional Engineering Publishing, 1998, p.370.
23. Avello A, Jimenez JM, Bayo E, et al. A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformation. *Comput Meth Appl Mech Eng* 1993; 107(3): 313–339.
24. Haug EJ. *Computer aided kinematics and dynamics of mechanical systems. Volume I: basic methods*. Needham Heights, MA: Allyn and Bacon, 1989, p.500.
25. Garcia de Jalon J and Bayo E. *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. New York: Springer-Verlag, 1994.
26. Watton J. *Fluid power system: Modelling, simulation, analog and microcomputer control*. London: Prentice Hall International (UK) Ltd, 1989, p.490.
27. Handroos HM and Vilenius MJ. Flexible semi-empirical models for hydraulic flow control valves. *J Mech Des* 1991; 113(3): 232–238.
28. Amrmstrong-Helouvry B, Dupont P and De Wit CC. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 1994; 30: 1083–1138.
29. Blundell M and Harty D. *The multibody system approach to vehicle dynamics*. Oxford, UK, Burlington, MA: Elsevier Butterworth-Heinemann, 2004, p.518.
30. Ebrahimi S and Eberhard P. Aspects of contact problems in computational multibody dynamics. In: Carlos García Orden Juan, M Goicolea José and Javier Cuadrado (eds) *Multibody dynamics: computational methods and applications*. The Netherlands: Springer, 2007, pp.23–47.
31. Moisio S, Korkealaakso PM and Rouvinen AJ. Contact modeling in real-time simulation of an underground wheeled loader. In: *The 1st joint international conference on multibody system dynamics*, Lappeenranta, Finland, 25–27 May 2010, p.64.
32. Drumwright E. A fast and stable penalty method for rigid body simulation. *IEEE Trans Visual Comput Graphics* 2008; 14(1): 231–240.
33. Escalona JL, Gonzalez M, Zaazaa KE, et al. A technique for validating a multibody wheel/rail contact algorithm. In: *Proceeding of DETC'03. ASME 2003 design engineering technical conferences and computers and information in engineering conference*, Chicago, USA, 2003.
34. Anitescu M. A fixed time-step approach for multibody dynamic with contact and friction. In: *International conference on intelligent robots and system*, Las Vegas, NV, 27 October–1 November 2003, vol. 4, pp.3725–3731.
35. Johnson KL. *Contact mechanics*. Cambridge, UK: Cambridge University Press, 1987, pp.1–468.
36. Ebrahimi S, Hippmann G and Eberhard P. Extension of the polygonal contact model for flexible multibody systems. *Int J Appl Math Mech* 2005; 1: 33–50.
37. Rouvinen AJ, Lehtinen T and Korkealaakso PM. Container gantry crane simulator for operator training. *Proc IMechE Part K: J Multi-body Dynamics* 2005; 219: 325–336.
38. Drumwright E and Sheel DA. An evaluation of methods for modeling contact in multibody simulation. In: *IEEE international conference on robotics and automation*, Shanghai International Conference Center, Shanghai, China, 9–13 May 2011, pp.1695–1701.
39. Redon S, Kheddary A and Coquillart S. Fast continuous collision detection between rigid bodies. *Comp Graphics Forum* 2002; 21(3): 279–288.
40. Lai KC and Kang SC. Collision detection strategies for virtual construction simulation. *Automat Construct* 2009; 18(6): 724–736.
41. Zachmann G. Rapid collision detection by dynamically aligned DOP-trees. In: *Proceedings of the IEEE virtual reality annual international symposium*, Atlanta, GA, 14–18 March 1998, pp.90–97.
42. Gottschalk S, Lin MC and Manocha D. OBBTree: a hierarchical structure for rapid interference detection. In: *Computer graphics (SIGGRAPH '96 proceedings) proceedings* of the 23rd annual conference on *computer graphics*, New Orleans, LA, 4–9 August 1996, pp.171–180.
43. Muth B, Muller MK, Eberhard P, et al. Collision detection and administration methods for many particles with different sizes. In: *4th international conference on discrete element methods, DEM 2007*, Brisbane, Queensland, Australia, 27–29 August 2007.
44. Moore M and Wilhelms J. Collision detection and response for computer animation. *Comp Graphic* 1998; 22(4): 289–298.
45. Hippmann G. An algorithm for compliant contact between complexly shaped bodies. *Multibod Syst Dynam* 2004; 12: 345–362.
46. Hippmann G. An algorithm for compliant contact between complexly shaped surfaces in multibody dynamics. In: Jorge AC Ambrósio (ed.) *Multibody Dynamics IDMEC/IST*, Lisbon, Portugal, 1–4 July 2003, pp.345–362.
47. Hasegawa S and Sato M. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Eurographics* 2004; 23(3): 529–538.
48. Hasegawa S, Fujii N, Koike Y, et al. Real-time rigid body simulation based on volumetric penalty method. In: *Proceedings of the 11th symposium on haptic interfaces for virtual environment and teleoperator system (HAPTIC)*, Los Angeles, CA, 22–23 March 2003, pp.326–332.
49. Li H and Schindler C. Three-dimensional finite element and analytical modeling of tyre-soil interaction. *Proc IMechE Part K: J Multi-body Dynamics* 2013; 227(1): 42–60.
50. Sharaf AM, Rahnejat H and King PD. Analysis of handling characteristics of all-wheel-drive off-road vehicles. *Int J Heavy Vehicle Syst* 2008; 1: 89–106.
51. Canudas de Wit C, Astrom KJ and Lischinsky P. A new model for control of systems with friction. *IEEE Trans Automat Cont* 1995; 40(3): 419–425.
52. Canudas de Wit C, Tsiotras P, Velenis E, et al. Dynamics friction models for longitudinal road/tire interaction: experimental results. *Citeseer (cited 15 October 2012)*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.3250&rep=rep1&type=pdf (2002, 28 July 2013).

53. Canudas de Wit C, Tsiotras P, Velenis E, et al. Dynamic friction models for road/tire longitudinal interaction. *Vehicle Syst Dynam* 2003; 39: 189–226.
54. Canudas de Wit C and Tsiotras P. Dynamic tire friction models for vehicle traction control. In: *Proceedings of the 38th IEEE conference on decision and control*, Phoenix, AZ, USA, 7–10 December 1999; vol. 4, pp.3746–3751.
55. Do NB, Ferri AA and Bauchau OA. Efficient simulation of a dynamic system with LuGre friction. *J Comput Nonlinear Dynam* 2007; 2: 281.
56. Olsson H, Astrom KJ, Canudas de Wit C, et al. Friction models and friction compensation. *Eur J Control* 1998; 4: 176–195.
57. Korkealaakso PM, Rouvinen AJ, Moisio SM, et al. Development of real-time simulation environment. *Multibod Syst Dynam* 2007; 17: 177–194.

Appendix

Notation

a_1, a_2	radii of box A
b_1, b_2	radii of box B
A_1, A_2	normalized axes of box A
A_A	area behind the hydraulic piston
$\mathbf{A}_A$	rotation matrix of body A
A_B	area a front of hydraulic piston
$\mathbf{A}_B$	rotation matrix of body B
A_n	area of cylinder chamber
B_1, B_2	normalized axes of box B
B_{ei}	effective bulk modulus
B_j	bulk modulus of component j
B_k	bulk modulus of component k
B_n	body with number of *n*
B_{oil}	bulk modulus of oil
c	damping coefficient
C	damping factor
$\mathbf{C}$	total inertia
C_v	flow rate constant
$\mathbf{d}_n$	joint relative displacement vector
$\ddot{\mathbf{d}}_n$	joint relative acceleration
$\dot{\mathbf{d}}_n$	joint relative velocity
dp	pressure different between pressure ports
$\mathbf{E}_q$	velocity transformation matrix
$\mathbf{E}_z$	new velocity transformation matrix
$f(\dot{x})$	velocity-dependent friction coefficient
F_c	Coulomb friction force
F_{lmin}, F_{lmax}	hydraulic cylinder end damping
$\mathbf{F}_n$	external force at body *n*
F_n	normal force
$\mathbf{F}_q$	inverse velocity transformation matrix
F_s	normalized static friction
F_s	total force produced by the cylinder
F_μ	total friction force of the cylinder
$g(v)$	Stribeck sliding friction force effect
$\mathbf{I}$	identity matrix
$\mathbf{J}_n$	inertia of body *n*
k	end damping spring constant
K	coefficient of elasticity
l	cylinder maximum stroke
$\mathbf{L}$	normalized direction
$\mathbf{m}_n$	mass matrix of body *n*
$\mathbf{M}$	total mass matrix
$\mathbf{n}$	contact normal
n_c	number of hydraulic component
n_h	number of hoses
N_b	number of contiguous bodies
$\dot{p}_l$	hydraulic pressure
p_n	pressure acting in the cylinder chamber
$\dot{\mathbf{p}}_q$	dependent velocities
$\mathbf{p}_z$	new position vector
$\dot{\mathbf{p}}_z$	new velocity
$\ddot{\mathbf{q}}$	acceleration
$\mathbf{q}_n$	generalized coordinate of joint *n*
Q	flow over restrictor
$\mathbf{Q}$	total forces
Q_{jA}	flow rate behind the cylinder piston
Q_{jB}	flow rate a front of cylinder piston
Q_{ij}	outgoing or incoming flow rate
$\mathbf{r}_n$	relative kinematics of position for body *n*
$\dot{\mathbf{r}}_n$	relative kinematics of velocity for body *n*
$\ddot{\mathbf{r}}_n$	acceleration of body *n*
R	projection matrix
$\mathbf{s}_{n-1}$	vector from point *n*–1 to *n*–1’
$\mathbf{T}$	distance from centre
T	time
$\mathbf{T}_n$	torque at body *n*
$\bar{\mathbf{u}}_A$	relative position vector of body A
$\bar{\mathbf{u}}_B$	relative position vector of body B
$\mathbf{u}_n$	new shifted position to the centre of gravity of r_n
$\ddot{\mathbf{u}}_n$	acceleration of body body *n*
U	variable of spool or poppet position
$\dot{U}$	velocity of position displacement
U_{ref}	spool reference position
$\mathbf{v}_n$	velocity at joint *n*
v_s	Stribeck relative velocity
v_{AB}	dot product of relative velocity
V_i	total hydraulic volume
V_j	volume of single component
V_{jA}	volume behind the hydraulic piston
V_{jB}	volume a front of hydraulic piston
V_k	volume of single hose
x	displacement of cylinder stroke
$\dot{x}$	stroke velocity of the cylinder
z	deflection of the bristles
$\delta\boldsymbol{\omega}_n$	virtual angular velocities
$\delta\dot{\mathbf{u}}_n$	virtual translational
$\delta\dot{\mathbf{q}}$	dependent virtual velocities
ζ_0	distance from an undeformed bristle point to origin point
ζ_1	distance from the deformed bristle point to origin point
η	cylinder efficiency

σ_0	longitudinal lumped stiffness
σ_1	longitudinal lumped damping coefficient
σ_2	viscous relative damping
τ	time constant describing dynamic of valve spool
$\boldsymbol{\omega}_d$	relative angular velocity of the joint
$\dot{\boldsymbol{\omega}}_n$	relative angular acceleration of body n
$\dot{\boldsymbol{\omega}}_d$	relative angular acceleration of the joint
$\tilde{\boldsymbol{\omega}}_n$	skew-symmetric matrix of the cross-product
$\dot{\tilde{\boldsymbol{\omega}}}_{n-1}$	differentiating of skew-symmetric matrix of body n–1

Publication IV

Baharudin, E., Nokka J., Montonen J.-H., Immonen P., Rouvinen A., Laurila L., Lindh T., Mikkola A., Sopanen J., and Pyrhönen J.
Simulation Environment for Real-time Dynamic Analysis of Hybrid Mobile Machines

Proceedings of the ASME International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2-5, 2015, Boston, Massachusetts, USA

DETC2015-47024

SIMULATION ENVIRONMENT FOR THE REAL-TIME DYNAMIC ANALYSIS OF HYBRID MOBILE MACHINES

Ezral Baharudin
Lab. of Machine Design
Lappeenranta University of Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
ezral.baharudin@lut.fi

Jarkko Nokka
Lab.of Electrical Drives Tech.
Lappeenranta University of Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
jarkko.nokka@lut.fi

Henri Montonen
Lab. of Control Eng. & Digital Tech.
Lappeenranta University of
Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
henri.montonen@lut.fi

Paula Immonen
Lab.of Electrical Drives Tech.
Lappeenranta University of
Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
paula.immonen@lut.fi

Asko Rouvinen
Mevea Ltd.
Laserkatu 6
Lappeenranta, 53850, Finland
asko.rouvinen@mevea.com

Lasse Laurila
Lab.of Electrical Drives Tech.
Lappeenranta University of
Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
lasse.laurila@lut.fi

Tuomo Lindh
Lab. of Control Eng. & Digital Tech.
Lappeenranta University of
Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
tuomo.lindh@lut.fi

Aki Mikkola
Lab. of Machine Design
Lappeenranta University of
Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
aki.mikkola@lut.fi

Jussi Sopanen
Lab. of Machine Dynamics
Lappeenranta University of Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
jussi.sopanen@lut.fi

Juha Pyrhönen
Lab.of Electrical Drives Tech.
Lappeenranta University of Technology
Skinnarilankatu 34
53850 Lappeenranta, Finland
juha.pyrhönen@lut.fi

ABSTRACT

The interest in using hybrid technology in Non-Road Mobile Machinery (NRMM) has increased significantly in the late 2000s due to tightening emission regulations (Tier 4). In general, utilization of hybrid technology can simplify the vehicle driveline compared to conventional mechanical and hydraulic power transmissions. On the other hand, hybrid technology and its different driving modes and multiple power sources creates new challenges in the design process. Many industries have used co-simulation and virtual prototyping approaches successfully as a development and diagnostic tool. However, it is still rarely used in the design of hybrid mobile machines. This is due to the fact, that the computer analysis of a mobile machine is a multidisciplinary task which requires a deep knowledge in several engineering areas. In this paper, a novel real-time co-simulation platform is presented that couples multi-body dynamics based physics modelling and Matlab/Simulink –based hybrid driveline modelling. The presented approach enables a fast and accurate virtual prototyping tool to calculate dimension hybrid driveline components and test various hybridization concepts.

INTRODUCTION

The interest in hybridizing Non-Road Mobile Machinery (NRMM), such as mine loaders, straddle carriers, and harvesters, has recently increased significantly, mainly due to tightening emission regulations (Tier 4). Hybrid technology with multiple power sources creates new challenges in the design process. An employment of co-simulation and virtual prototyping approaches has rapidly become an essential development and diagnostic tool for a number of industries. However, it is still rarely used in the design of hybrid mobile machines. This is due to the fact, that the computer analysis of a mobile machine is a multidisciplinary task which requires a deep knowledge in several engineering areas and their interaction, in particular.

Multi-body system software (MBS) can be used in a co-simulation of control systems, actuators, and mechanics in a straightforward manner [1]. For example, Simpack multi-body software can be used with Matlab/Simulink to simulate active steering systems of vehicles [2]. The model of the vehicle running in Simpack and Matlab/Simulink is used for implementation by the controller. Ping He et al. studied integration of semi-active suspension and vehicle dynamics control [3]. They ran a multi-body model of a vehicle in MSC.ADAMS software and used Matlab/Simulink to test different controllers. Mousseau et al. presented a comprehensive vehicle dynamics model for simulating the dynamic response of a ground vehicle on rough surfaces [4]. They used multi-body system simulation (MSC ADAMS) to simulate the vehicle, a nonlinear finite element (FE) program for the tires, and an interface to transfer the data between simulations.

The objective of this paper is to introduce a detailed simulation environment based on the multi-body dynamics approach. The use of modern multibody simulation techniques enables the description of complex products such as mobile machinery with a high level of detail while still solving the equations of motion in real-time. This technology has been utilized in user training and more recently, in product development. For product development, real-time simulation makes it possible to account for the machine users and their needs early on in the concept development phase.

The simulator environment presented in this paper is tailor designed for analysis of hybrid vehicles. This environment allows a detailed multi-body model of a mobile machine to be run in real time in a realistic virtual environment by a human driver. By coupling to simulation models of electric drives in an another software, for example Matlab/Simulink, the whole system simulation environment also allows a number of design features to be studied, including fuel consumption, control strategies, and optimization of hybrid vehicles. The introduced implicit co-simulation environment consists of modules such as mechanisms, tires, hydraulic actuators, and various other models as well as the externally coupled hybrid driveline simulation and its components in Simulink, and the whole system can be solved in real-time.

MULTI-BODY SYSTEM MODELING

In this study, an open loop semi recursive method is used in multibody modelling. In this method, kinematic properties such as position, velocity, and acceleration are developed based on the relative coordinates between neighboring bodies connected by a joint [5]. This algorithm has been used and extended by several researchers and been generalized to improve its implementation and efficiency [6][7]. The dynamics equation in a recursive formulation is written in terms of the degrees of freedom of the system and, typically, is written with a lower dimensionality than those in an augmented formulation.

A schematic representation of a multibody system is depicted in Fig. 1. Each body number of the multibody system can be represented as B_n. Preceding bodies are designated B_{n-1}. The arranged set of body numbers is called a “body connection array” [8]. Joint numbers are named before body numbers. The revolute, prismatic, cylindrical, translational, or spherical joints can consist of single or multiple axes. In this study, the multibody implementation is general and it also supports rheonomic constraints.

Kinematics

Relative motion between neighboring bodies and constraints are the two main aspects of recursive kinematics used to generate the total system matrices and solve the equations of motion for the multi-body system. Fig. 1 shows an elementary system of two bodies interconnected by a joint.

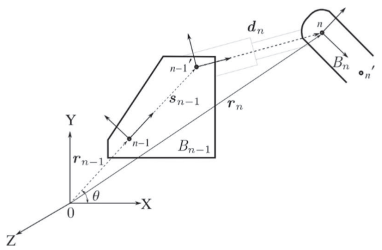


FIGURE 1. RELATIONSHIP OF NEIGHBORING BODIES.

The orientation relationship of contiguous bodies is obtained by sequentially transforming from the body reference frame on body B_0 to the body reference frame on body B_n. The system of bodies is considered an open chain, and point 0 is the global reference frame. The relative kinematics of position $\boldsymbol{r}_n$ for each body can be described with respect to the global reference as follows [9].

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} + \boldsymbol{s}_{n-1} + \boldsymbol{d}_n \,, \tag{1}$$

where $\boldsymbol{d}_\mathrm{n}$ is the joint relative displacement vector from point n-1 to point n , and $\boldsymbol{s}_{n-1}$ is a multiplication of rotation matrix $\mathbf{A}_{n-1}$ with body fixed frame vector $\bar{\boldsymbol{s}}_{n-1}$. In this study, Euler parameter and a normalized constraint are enforced by employing a direct correction approach. The velocities can be determined from the time derivative of equation (1).

$$\boldsymbol{v}_n = \dot{\boldsymbol{r}}_{n-1} + \widetilde{\boldsymbol{\omega}}_{n-1}\boldsymbol{s}_{n-1} + \dot{\boldsymbol{d}}_n \,, \tag{2}$$

$$\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} + \boldsymbol{\omega}_\mathrm{d} \,, \tag{3}$$

where matrix $\widetilde{\boldsymbol{\omega}}_{\boldsymbol{n-1}}$ is the skew-symmetric matrix representation of angular velocity vector. Vector $\boldsymbol{\omega}_\mathrm{d}$ is the relative angular velocity of the joint. The acceleration equations can be obtained by differentiating equations (2) and (3) as follows

$$\ddot{\boldsymbol{r}}_n = \ddot{\boldsymbol{r}}_{n-1} + \dot{\tilde{\boldsymbol{\omega}}}_{n-1}\boldsymbol{s}_{n-1} \\ +\tilde{\boldsymbol{\omega}}_{n-1}\tilde{\boldsymbol{\omega}}_{n-1}\boldsymbol{s}_{n-1} + \ddot{\boldsymbol{d}}_n \tag{4}$$

$$\dot{\boldsymbol{\omega}}_n = \dot{\boldsymbol{\omega}}_{n-1} + \dot{\boldsymbol{\omega}}_{\mathrm{d}}. \tag{5}$$

In equations (1) - (5), the terms $\boldsymbol{d}_n$and $\boldsymbol{\omega}_{\mathrm{d}}$ are functions of relative joint position and joint angular velocity, respectively. The functions of velocity and acceleration for each common type of joint (prismatic, spherical, *etc*.) can be obtained by deriving expressions as suggested by Avello et al. [10].

Equation of motion

The equations of motion for open loop system, which are composed of N_{b} bodies, interconnected by kinematic joints can be written as

$$\delta\boldsymbol{V}^{*\mathrm{T}}\left(\mathbf{M}\dot{\boldsymbol{V}} + \boldsymbol{C} - \boldsymbol{Q}\right) = \mathbf{0}, \tag{6}$$

where $\delta\boldsymbol{V}^*$ is $6N_{\mathrm{b}}$ of virtual Cartesian velocities vector, $\mathbf{M}$ is $6N_{\mathrm{b}}$ x $6N_{\mathrm{b}}$ of mass matrix, $\dot{\boldsymbol{V}}$ is $6N_{\mathrm{b}}$ of Cartesian accelerations vector, $\boldsymbol{C}$ is $6N_{\mathrm{b}}$ of centrifugal forces vector and $\boldsymbol{Q}$ is $6N_{\mathrm{b}}$ external forces vector. All terms can be written as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & & \\ & \ddots & \\ & & \mathbf{M}_{N_{\mathrm{b}}} \end{bmatrix}, \\ \boldsymbol{C} = \begin{Bmatrix} \boldsymbol{C}_1 \\ \vdots \\ \boldsymbol{C}_{N_{\mathrm{b}}} \end{Bmatrix}, \\ \boldsymbol{Q} = \begin{Bmatrix} \boldsymbol{Q}_1 \\ \vdots \\ \boldsymbol{Q}_{N_{\mathrm{b}}} \end{Bmatrix}, \\ \dot{\boldsymbol{V}} = \begin{Bmatrix} \dot{\boldsymbol{V}}_1 \\ \vdots \\ \dot{\boldsymbol{V}}_{N_{\mathrm{b}}} \end{Bmatrix} \tag{7}$$

The terms in equation (7) can be decomposed into dedicated component as

$$\mathbf{M}_n = \begin{bmatrix} \mathrm{m}_n\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_n \end{bmatrix}, \\ \boldsymbol{C}_n = \begin{Bmatrix} \mathbf{0} \\ \tilde{\boldsymbol{\omega}}_n\mathbf{J}_n\boldsymbol{\omega}_n \end{Bmatrix}, \\ \boldsymbol{Q}_n = \begin{Bmatrix} \boldsymbol{f}_n \\ \boldsymbol{n}_n \end{Bmatrix}, \\ \boldsymbol{V}_n = \begin{Bmatrix} \ddot{\boldsymbol{g}}_n \\ \dot{\boldsymbol{\omega}}_n \end{Bmatrix} \tag{8}$$

where $\mathbf{M}_n$ is a 6x6 mass matrix, $\boldsymbol{C}_n$ and $\boldsymbol{Q}_n$ are 6x1 vectors corresponding to body n. m_i is the mass, $\mathbf{I}$ is the identity matrix with a size of 3x3, $\boldsymbol{\omega}_n$ is the angular velocity, $\boldsymbol{g}_n$ is the center of mass position, $\mathbf{J}_n$ is the 3x3 inertia tensor, $\boldsymbol{f}_n$ is 3x1 external forces vector, $\boldsymbol{n}_n$ is 3x1 external torque vector. The constant inertia tensor $\mathbf{J}_n$ can be computed as

$$\mathbf{J}_n = \mathbf{A}_n\bar{\mathbf{J}}_n\mathbf{A}_n^T. \tag{9}$$

The virtual velocities in equation (6) needs to be eliminated from the equation by expressing $\boldsymbol{V}^*$ in terms of the independent joint variables $\boldsymbol{z}$ as describe in [10] and [11]. With an assumption that the constraints are independent of time, linear relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{V}$ can be derive as

$$\boldsymbol{V} = \mathbf{R}\dot{\boldsymbol{q}}, \tag{10}$$

where $\mathbf{R}$ is $6N_{\mathrm{b}}\mathrm{x}F$ the velocity transformation matrix and F is number of degree of freedom. Decompose equation (10) into body-n become as

$$\boldsymbol{V}_n = \mathbf{R}_n\dot{\boldsymbol{q}}, \tag{11}$$

where $\mathbf{R}_n$ is velocity transformation matrix of body-n with 6xF matrix. By differentiating equation 10, the following result is obtained

$$\dot{\boldsymbol{V}} = \mathbf{R}\ddot{\boldsymbol{q}} + \dot{\mathbf{R}}\dot{\boldsymbol{q}} \tag{12}$$

Substituting equations (12) and (10) into equation (6), we could obtain a final equation of motion for the open loop system as

$$\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\boldsymbol{q}} = \mathbf{R}^{\mathrm{T}}(\boldsymbol{Q} - \boldsymbol{C}) - \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\boldsymbol{q}}. \tag{13}$$

To solve the closed loop systems, the systems are described with help of an augmented Lagrangian approach.

Modelling of hydraulics

In this study hydraulic actuators, the assumed drivers, are modelled using the lumped fluid theory, which divides a hydraulic circuit into discrete volumes with hydraulic pressure distributed equally. Valves can be modelled using a semi-empirical approach that makes use of flow parameters that can be obtained, in many cases, directly from manufacturer's catalogues. Usually, a hydraulic system has a high nominal frequency response, and therefore, the calculation time step should be selected carefully not beyond the edge point of numerical stability of hydraulic circuit. In this approach, pressure wave propagation in pipes and hoses is assumed to be negligible [12]. The hydraulic pressure in each hydraulic volume i can be described as follows

$$\dot{p}_{\mathrm{i}} = \frac{B_{\mathrm{ei}}}{V_{\mathrm{i}}}\sum_{j=1}^{n_{\mathrm{c}}} Q_{\mathrm{ij}}, \tag{14}$$

where B_{ei} is the effective bulk modulus that defines the flexibility of the hydraulics, Q_{ij} is the outgoing or incoming flow rate, and n_{c} is the total number of hydraulic components, all related to volume i, V_{i}. The effective bulk modulus can be calculated as

$$B_{\mathrm{ei}} = \frac{1}{\frac{1}{B_{\mathrm{oil}}} + \sum_{j=1}^{n_{\mathrm{c}}} \frac{V_{\mathrm{j}}}{V_{\mathrm{i}} B_{\mathrm{j}}} + \sum_{k=1}^{n_{\mathrm{h}}} Q_{\mathrm{ij}} \frac{V_{\mathrm{k}}}{V_{\mathrm{i}} B_{\mathrm{k}}}} \tag{15}$$

In equation (15), n_{h} is the total number of pipes and hoses related to volume i. The bulk modulus of oil, B_{oil}, accounts for the amount of non-dissolving air in the oil. It is a function of pressure, and the maximum value is typically B_{oilmax} = 1.6 GPa. The bulk modulus B_{j} and B_{k} of component j and k is also dependent on the component type.

For modelling purposes, a valve is assumed to consist of several adjustable restrictor valves, which can each be modelled separately [13]. With small pressure differences (< 1 bar), the flow over the restrictor is assumed to be laminar, and with larger differences, it is thought to be turbulent [14].

The force produced by the hydraulic cylinder F_{s} can be defined as

$$F_{\mathrm{s}} = p_1 A_1 - p_2 A_2 - F_{\mu} \,, \tag{16}$$

where F_{μ} is the total friction force of the cylinder, and p_1 and p_2 are pressures acting in the cylinder chambers. Friction force is a function of pressures, cylinder efficiency η , and velocity [24].

Modelling of tires

A number of dynamic friction models have been introduced in literature aiming to capture the transient behavior of the tire-road contact forces under time varying velocity conditions [15]. Dynamic models can be categorized based on the friction contact description. Dynamic models are called lumped if the friction model assumes a point tire-road friction contact as shown in Fig. 2. In the case of a lumped assumption, the mathematical model describing such a friction can be written in the form of ordinary differential equations that can be solved by a time integration scheme. An alternative to the lumped approach is distributed friction models that account for the existence of a contact patch between the tire and the ground. Distributed friction models can be computationally expensive due to the fact that friction models must be solved in terms of both time and space. The model parameters can be selected from the literature in such a way that the realistic behavior of the vehicle can be obtained. However, avoiding values that lead to a frequency higher than 150 Hz is recommended

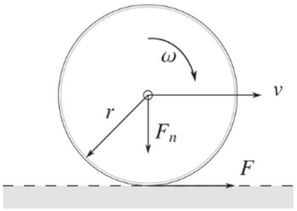


FIGURE 2. LUMPED LUGRE TIRE MODEL.

The LuGre tire model is a commonly used approach and it is based on the description of an elastic bristle at the microscopic level. At the contact point, the tangential force will affect the bristles by deflecting similar to springs, and create the friction force [16][17]. When a large tangential force is imposed, the bristles deflect excessively and start to slip. A concept of bristle deformation is depicted in Fig. 3.

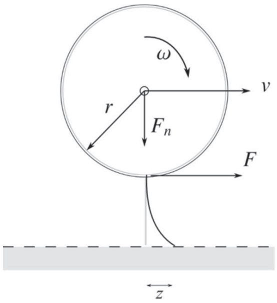


FIGURE 3. A BRISTLE DEFORMATION DURING AN ACCELERATION.

The LuGre friction model can be seen as an enhancement of the Dahl model as its accounts for the Stribeck effect. With this approach, many important aspects of friction including stiction, the Stribeck effect, stick slip, zero slip displacement, and hysteresis can be accounted for. When the tire rotates, the bristle will deflect with respect to time and the deflection of the bristles denoted by z can be evaluated as follows

$$\dot{z} = v_{\mathrm{r}} - \frac{\sigma_0 |v_{\mathrm{r}}|}{g} z \,, \tag{17}$$

where g is friction, σ_0 is the rubber longitudinal lumped stiffness and v_{r} is the relative velocity between two sliding surfaces defined as follows (see also Fig. 3)

$$v_{\mathrm{r}} = r\omega - v \,, \tag{18}$$

Friction g needed in equation (17) can be written as

$$g = \mu_{\mathrm{c}} + (\mu_{\mathrm{s}} - \mu_{\mathrm{c}}) e^{-\left|\frac{v_{\mathrm{r}}}{v_{\mathrm{s}}}\right|^{-\frac{1}{2}}} \,, \tag{19}$$

where μ_{c} is normalized Coulomb friction, μ_{s} normalized static friction and v_{s} is the Stribeck relative velocity. Friction, g must always be positive, and it depends on the material properties, lubricants, temperature, and other factors. Friction force based on the bristle deformation can be expressed as follows

$$F_{\mathrm{s}} = (\sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v_{\mathrm{r}}) F_{\mathrm{n}}\,, \tag{20}$$

where σ_1 is the longitudinal lumped damping coefficient, σ_2 is viscous relative damping and F_{n} is the normal force.

To obtain a computationally efficient and accurate tire model, a tire can be presumed to be a series of discs, as shown in Fig. 4, based on the width of the tire. In this approach, disks can be assumed to be simple rigid bodies, whereas the profile of the ground can be constructed depending on the required environment (off-road profile). In the Lugre tire implementation, all forces are applied to the hub of the tire, denoted as A in Fig. 4.

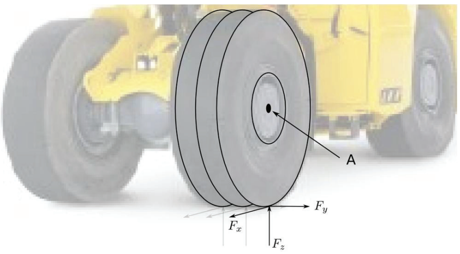


FIGURE 4. AN IMPLEMENTATION OF A TIRE MODEL.

In this study, the tire and hydraulics are coupled with a multibody model using a force connection i.e. forces produced by hydraulics and tires are evaluated in their own sub-models.

MODELLING OF THE VEHICLE DRIVELINE

Simulation of a mobile working machine, involving mechanical components, hydraulics, tires, electric drives as well as active electronic control devices, can be accomplished by combining tools that deal with the simulation of the different subsystems. This can be accomplished, in practise, by coupling multi-body simulation software with external simulation environment of electrical components.

A hybrid working machine model is prepared by combining all the component models together. In this way, a modular model is obtained, and it will be easy to modify the model by changing just one component inside the entire model, based on the needs of different machines and different hybridization principles.

Electric drive model

The electric drive can be modeled in several different ways. Depending on the aim of the simulation of hybrid working machines and the level of accuracy, there is a possibility to use detailed dynamic models or quasi-static models. A Quasi-static model of an electrical machine is commonly used in in the modelling and analysis of electro-hybrid vehicles at a system level [18], [19]. The quasi-static model is very close to the dynamic model with an accurate value of the energy consumption [20]. In the quasi-static model, the electric motor is modelled in a straightforward manner by only using a first- or second-order time constant model. The faster electrical dynamics are neglected and the quasi-static model only keeps the main dynamics. The torque of an electric machine is assumed to follow the torque reference limited by the maximum torque curve and delayed by the time constant filter. The losses in the electric machine are taken into account using efficiency maps [21]. The dynamic model is particularly used to study the control of a system [20]. With the dynamic model the converter model can be included, and the control of the electric machine can be designed and tested.

The electric drive here is modeled in Matlab Simulink. The main components typically include the traction motor, frequency converter, and energy storage. Here, a permanent magnet synchronous machine (PMSM) is selected for the traction motor. The PMSM is well suited for hybrid electric vehicles and hybrid working machines due to the high overall efficiency within a large range of the torque-speed plane, high overload capacity, and its small installation space and weight [22]. The PMSM is modelled here by applying the space vector theory on the rotor co-ordinate system.

The stator voltage equations of the PMSM in the rotor co-ordinate system are [23]

$$\begin{aligned} u_{\mathrm{d}} &= R_{\mathrm{s}} i_{\mathrm{d}} + \dot{\Psi}_{\mathrm{d}} - \omega \Psi_{\mathrm{q}} \\ u_{\mathrm{q}} &= R_{\mathrm{s}} i_{\mathrm{q}} + \dot{\Psi}_{\mathrm{q}} + \omega \Psi_{\mathrm{d}}\,, \end{aligned} \tag{21}$$

where the flux linkages are

$$\begin{aligned} \Psi_{\mathrm{d}} &= L_{\mathrm{sd}} i_{\mathrm{d}} + \Psi_{\mathrm{PM}} \\ \Psi_{\mathrm{q}} &= L_{\mathrm{sq}} i_{\mathrm{q}}\,. \end{aligned} \tag{22}$$

In the above equations (21-22), u refers to voltage, R to resistance, i to current Ψ to flux linkage and L is inductance – subscripts refer to either d, q or permanent magnet (PM) components. The stator currents are solved based on the voltage and flux linkage equations. The electromagnetic torque T_{em} is calculated by the permanent magnet flux Ψ_{PM}, stator currents and inductances $(L_{\mathrm{sd}}, L_{\mathrm{sq}})$ as

$$T_{\mathrm{em}} = \frac{3p}{2}\left[\Psi_{\mathrm{PM}} i_{\mathrm{q}} + \left(L_{\mathrm{sd}} - L_{\mathrm{sq}}\right) i_{\mathrm{d}} i_{\mathrm{q}}\right]. \tag{23}$$

A two-axis equivalent circuit of undampened PMSM based on equations (21) - (23) is presented in Fig. 5.

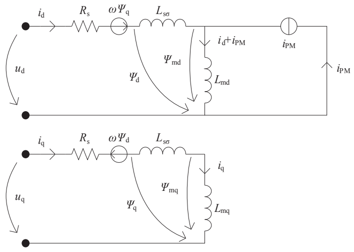


FIGURE 5. TWO-AXIS EQUIVALENT CIRCUIT OF UNDAMPENED PMSM.

The frequency converter is modeled purely as a current control circuit, and the switching stage of the inverter is not simulated. This reduces the overall accuracy of the inverter model, but increases the overall simulation performance as real-time simulation is needed. In addition, when combined with a two-axis model of an electrical motor, the added accuracy is negligible. The inverter current control circuit controls the voltage of the motor by a current error based closed loop control.

The electrical energy storage in this case is a battery. The battery is modeled simply with static voltage and static internal resistance according to Fig. 6 [24].

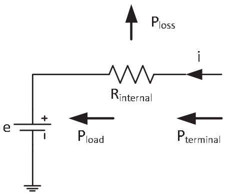


FIGURE 6. BATTERY MODEL. THE TERMINAL POWER P_{terminal} IS DIVIDED INTO LOSS COMPONENT P_{loss} OF THE INTERNAL RESISTANCE R_{internal} AND ACTUAL LOAD POWER P_{load}.

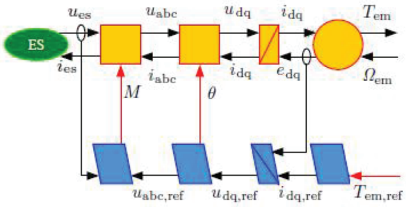


FIGURE 7. MODEL OF A PMSM DRIVE ACCORDING TO THE EMR. ES IS THE ENERGY SOURCE TO THE SYSTEM, u_{SDQ} IS THE STATOR VOLTAGE, i_{SDQ} THE STATOR CURRENT, e_{SDQ} THE STATOR ELECTROMOTIVE FORCE, T_{EM} ELECTROMAGNETIC TORQUE AND Ω_{EM} MECHANICAL ANGULAR VELOCITY.

The modelling of the electric drive is based on an energetic macroscopic representation (EMR), which is an energy flow-based graphical modelling tool to illustrate the energy flow in a complex electromechanical system. Fig. 7 illustrates the electric drive model according to the EMR [25], [26].

The input values of the electric drive model are reference torque $T_{\mathrm{em,ref}}$ and mechanical angular velocity Ω_{em} and the output value is the electromagnetic torque. The value of $T_{\mathrm{em,ref}}$ is determined by the position of the drive pedal and the mechanical angular velocity Ω_{em} is determined from the mechanics simulation system, which is driven with an electric motor.

Diesel engine model

The Diesel generator model is based on a known static efficiency map of a diesel engine combined with a transient fuel consumption model made according to the research by Lindgren (2005) as well as Lindgren and Hansson (2004) [27],[28]. The model calculates the diesel oil consumption based on the efficiency map information, presented in Fig. 8, as a function of rotational speed and torque and corrects the result with a so-called transient factor, which takes the time derivative (movement) of the operational point into account. The torque production is simply based on a 1st order reference follower with the time constants of an actual machine, combined with maximum torque limits.

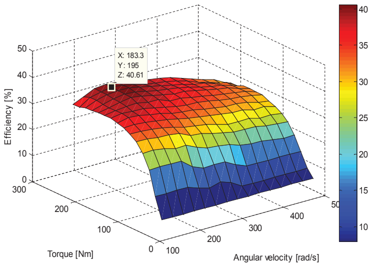


FIGURE 8. EFFICIENCY MAP OF A DIESEL ENGINE AS A FUNCTION OF ANGULAR VELOCITY AND TORQUE. HIGHLIGHTED IS THE MAXIMUM EFFICIENCY OPERATIONAL POINT.

SIMULATION SYSTEM COMPONENTS

The minimum simulation environment consists of the multi-body model of a simulated mobile working machine, the environment model, and the solver program. To increase realism, additional computers can be connected to the simulator PC to calculate a view from, for example, the driver's seat of the machine. This view can be shown using projectors or displays. Fig. 9 shows the simulator setup at LUT.

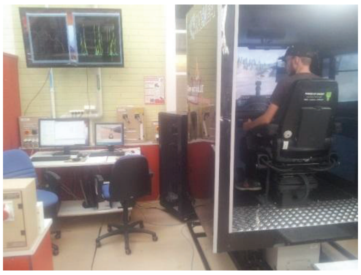


FIGURE 9. VIRTUAL SIMULATOR CABIN ENABLES INTERACTIVE SIMULATION WITH A HUMAN OPERATOR.

The virtual simulator cabin at LUT is equipped with 6 LCD displays, to project view forwards, left, right, up and down from the driver's seat. The cabin is equipped with a motion platform. The solver program calculates the forces affecting the driver and the cabin emulates the stimulated movement. The motion platform is driven by electrical servo motor powered linear actuators.

The solver program has an I/O interface for connecting physical instruments such as ECUs, joysticks, pedals, meters, etc. In this case, the human interface devices (joysticks and pedals) are connected using this I/O interface.

The external interface used in the co-simulation is built around a TCP Socket connection. The synchronization is done by a Mevea Solver and uses the socket ability to wait for the read buffer to be filled before continuing.

The external interface also makes it possible to attach separate applications to the simulation loop. The real time simulation data can be used as sensor data in the external application. For example, data such as magnitudes of forces, speeds, and angles can be sent to the external application. This allows parts, such as the motor control, to be made in the external application. The connection between the solver and the external application is established via a socket connection. The socket connection also makes it possible to attach hardware to the simulation loop, and replace the virtual parts of the simulation with actual parts.

When an external control application is used, the simulator software takes care of the timing. The simulator software sends a set of data to the external application, when the calculation is complete a set of return values is sent back to the simulator. New values are then updated to the multi-body dynamics. For example a mechanical driveline component has an angular velocity and a load torque. These values are sent from the simulation software to the external application. External applications can then write either the angular velocity or torque of the component.

CASE EXAMPLE

As a case example, a hybrid heavy duty working machine - underground mine loader- is modelled in its natural working environment.

Principles and model layouts

The hybridization of the case example is here implemented with a series hybrid system, where the diesel engine is totally disconnected from the mechanics. All the power of all mine loader functions is delivered electrically and transferred to either hydraulic or mechanical drivelines.

In the series hybrid version, the diesel engine is utilized merely as a component of a diesel generator set, providing electrical power to the introduced new components - electrical drives or to the energy storage. Otherwise the system is the same as in the original, a purely diesel-powered mine loader. In the driveline system, the traction is implemented by using a gearbox, a reverser, a center differential, a front and rear differential and a planetary gear in every wheel. The hydraulic system consists of two variable displacement pumps (steering pump and bucket pump), two directional valves and cylinders (steering cylinders, lift cylinder and tilt cylinder). Fig. 10 shows the schematic diagram of the transmission and Fig. 11 is a diagram of the hydraulic system.

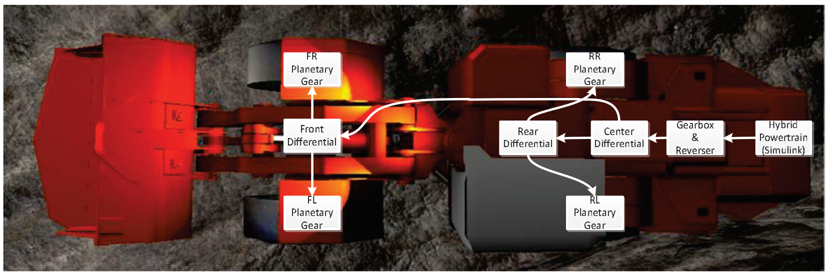


FIGURE 10. SCHEMATIC DIAGRAM OF THE MINE LOADER TRACTIVE DRIVELINE.

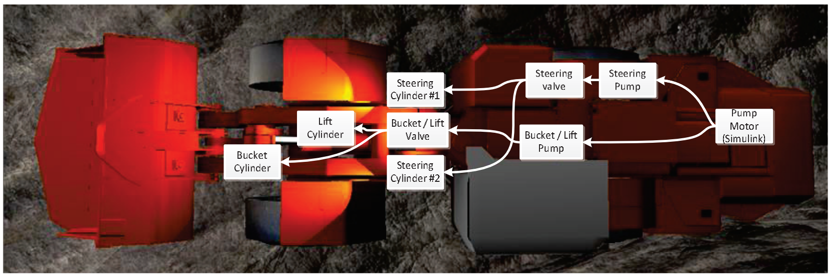


FIGURE 11. SCHEMATIC DIAGRAM OF THE MINE LOADER HYDRAULIC SYSTEM.

The underground mine loader was modelled by coupling a Mevea real-time simulation environment, which is based on multi-body dynamics, and a Matlab/Simulink. The Mevea simulation environment is responsible for the mechanics and hydraulic simulation, while the electric drive, energy storage, and diesel generator set are modelled in Matlab/Simulink. The traction drive model in Simulink receives the reference torque from the position of the drive pedal from the Mevea simulation and the electric motor torque is the input of mechanics simulation. The Mevea simulation returns the angular velocity of the axle of the electric motor to Simulink. Fig. 12 shows the communication between the Mevea simulation environment and the Matlab/Simulink. The hydraulic motor drive generates the angular velocity signal for the Mevea simulation, which produces a load torque signal as feedback.

Multi-body Simulation models of working machines can be utilized when designing the hybridization of working machines. A fixed time step with a 1.2 ms explicit Runge-Kutta method is used in this model. With the simulation model, pressure levels and flow rates of work hydraulic actuators and torque and rotation speed of different parts of drive transmission, can be analyzed. The suitability of different hybrid drive systems and the fuel consumption of working machine can be researched through simulations.

Simulations can also be used to analyze the effect of different hybrid system component's sizing (diesel engine, electric machine, energy storage) as regards fuel consumption. The upper level control of the hybrid system also has a significant impact on the energy consumption; for example in the form of choosing the suitable operational point of the diesel engine or determining the operation of the pump drives. With simulation it is also possible to study how the different control strategies affect the energy consumption and performance of a working machine.

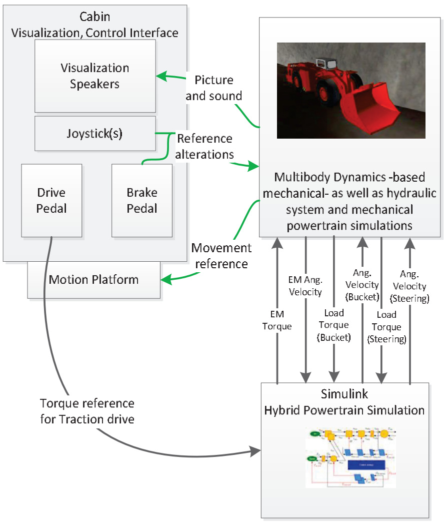


FIGURE 12. COMMUNICATION BETWEEN THE MEVEA REAL-TIME SIMULATION ENVIRONMENT AND THE MATLAB SIMULINK.

Simulink Models

The Simulink model consists of one tractive motor which produces torque to the mechanical driveline presented in Fig. 12 and two additional motors, which rotate the hydraulic pumps. The highest level simulation model is presented in Fig.13.

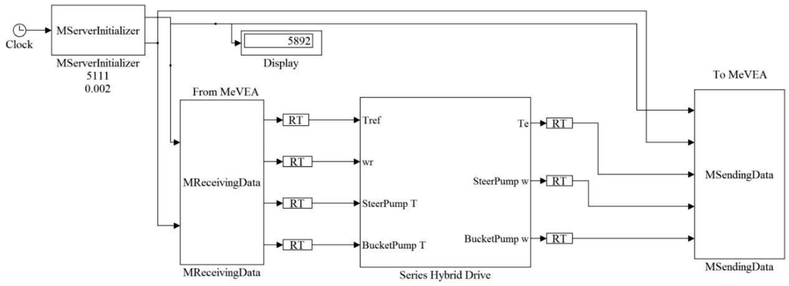


FIGURE 13. TOP LEVEL OF HYBRID DRIVELINE SIMULINK MODEL: SERVER INITIALIZER, DATA RECEIVE AND DATA SENDING BLOCKS FORMING THE MEVEA SIMULATION DATA INTERFACE, AND SERIES HYBRID DRIVELINE SUBSYSTEM CONTAINING THE COMPLETE HYBRID DRIVELINE WITH THE ASSOCIATED SUB-COMPONENTS LIKE GENSETS AND ELECTRICAL DRIVES.

Fig. 13 illustrates the fundamental structure of the Simulink model as well as the transfer layer construction. Once in every 2 ms the Simulink model receives updated actual values from the Mevea simulator, which in turn, simultaneously sends the updated values back. The series hybrid model then continues to calculate the next 2 ms cycle with 100 µs intervals. The transferred signals consist of mechanical driveline shaft signals, pump shaft signals and a traction motor control signal. The Hybrid driveline is further presented in Fig. 14.

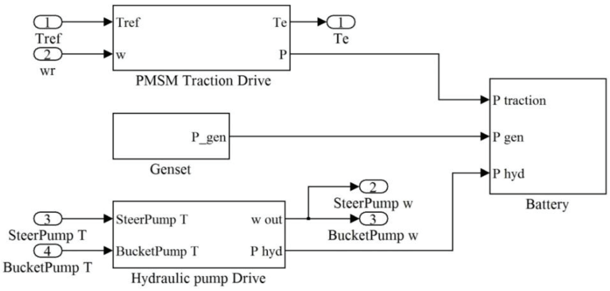


FIGURE 14. HYBRID DRIVELINE MODELLED IN SIMULINK. The MODEL CONSISTS OF ONE TRACTIVE, TORQUE CONTROLLED ELECTRICAL DRIVE, TWO SPEED CONTROLLED PUMP MOTOR DRIVES, a DIESEL GENERATOR SET AND ENERGY STORAGE (BATTERY).

The electrical drive models in Fig. 14 are all PMSM-based models, based on equations (21) – (23). The battery is modelled as a constant voltage, and a constant resistance storage, as presented in Fig. 6, and the diesel model consists of a 1st order angular velocity –dependent time constant torque follower, combined with an efficiency map and transient fuel consumption maps for fuel efficiency analysis.

Hydraulic Circuits

The hydraulic circuit includes two hydraulic pumps. The first pump supplies the steering cylinders and the second one the lift and bucket cylinders. Fig. 15 shows the simplified hydraulic schematic of the mine loader.

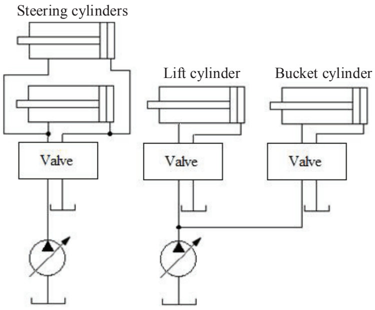


FIGURE 15. SIMPLIFIED HYDRAULIC SCHEMATIC OF THE MINE LOADER HYDRAULIC SYSTEM.

Simulation Results

A demonstrative work cycle was driven with the case example. The cycle consisted of driving the loader along the mineshaft to a pile of rocks, scooping them up and reversing all the way back to the starting area's dump location. From the Simulink model the key data used with hybridization – powers, battery state-of-charge (SoC), and fuel consumption were logged and are presented in Fig. 16 to Fig. 18. Additionally, the performance-related loop time and time step data were logged from the Mevea software and is presented in Fig. 19.

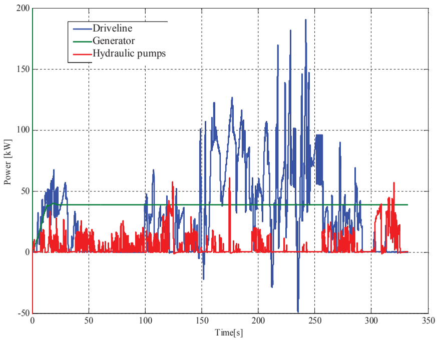


FIGURE 16. POWER VARIATIONS OF THE EXAMPLE WORK CYCLE. TRACTIVE POWER (BLUE) AND HYDRAULIC PUMP DRIVE (RED) DISCHARGE OFTHE BATTERY WHILST THE GREEN GENERATOR POWER CHARGES IT. IN ADDITION, WHEN THE TRACTIVE POWER GOES INTO NEGATIVE FIGURES, IT ALSO RECHARGES THE BATTERY.

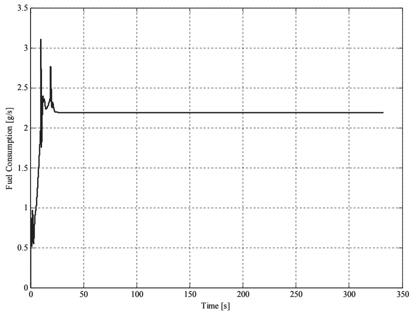


FIGURE 17. FUEL CONSUMPTION OF THE EXAMPLE WORK CYCLE. THE DIESEL GENERATOR OPERATES AT A FIXED OPERATING POINT PROVIDING STATIC POWER. AFTER THE INITIAL ACCELERATION OF THE GENSET THE FUEL CONSUMPTION STABILIZES ITSELF AT APPROXIMATELY 2.2 GRAMS PER SECOND.

From Fig. 16, the fundamental idea of series hybridization can be seen. While the hydraulic drive and tractive drive have quite fast dynamics in them, the diesel-powered generator operates at static power. The battery can be considered as a low-pass filter for the power, enabling the diesel generator set to operate constantly at high efficiency.

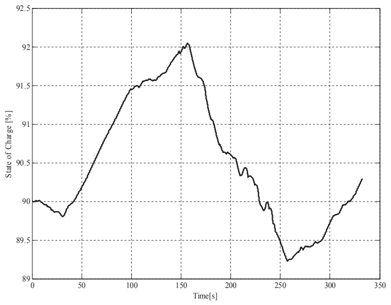


FIGURE 18. STATE OF CHARGE (SOC) GRAPH FROM THE EXAMPLE WORK CYCLE. AT FIRST, WHEN DRIVING DOWNHILL, THE POWER BALANCE IS POSITIVE AND THE BATTERY CHARGES UP. WHEN THE MACHINE REVERSES BACK UP, IT REQUIRES LARGE AMOUNTS OF POWER AND DEPLETES THE BATTERY QUITE RAPIDLY. AT THE END THE SOC IS APPROXIMATELY AT THE SAME POINT AS AT THE START.

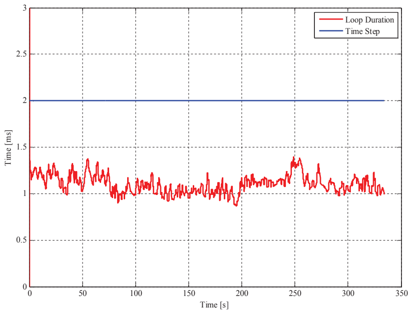


FIGURE 19. LOOP DURATION AND TIME STEP LENGTH LOGGED FROM AN EXAMPLE WORK CYCLE. LOOP DURATION REMAINS UNDER THE TIME STEP VALUE FOR THE WHOLE LENGTH OF THE SIMULATION. THIS MEANS THAT THE SIMULATION RUNS IN REAL-TIME.

CONCLUSIONS

A multi-body simulation based development environment for hybrid working machines was introduced. It was demonstrated that real-time simulation can be realized even with such a complex system that coupled multi-body simulation models of a mobile working machine and its virtual environment to electrical drive models, including interactivity with human operators. In the case example – the underground mine loader - the simulated signals of torque, speed, and pressures were at the right level and the driving feel was realistic enough to continue development and utilization of the multi-body simulation based development environment.

REFERENCES

[1] Gonzalez, F., Gonzalez, M., and Mikkola, A., 2010, "Efficient Coupling of Multibody Software with Numerical Computing Environments and Block Diagram Simulators", Multibody System Dynamics, 24(3), pp. 237-253.

[2] Dorsch, V., 2012, "Implementation of active Steering Systems into a Multi-body Vehicle Model by Co-Simulation". Mechatronics-REM.

[3] Ping He, Yiqi Wang, Yunging Zhang, Yang liu, Youzhong Xu , 2010, "Integrated control of semi-active suspension and vehicle dynamics control system", 2010 International Conference on Computer Application and System Modeling (ICCASM).

[4] Mousseau, C.W., Laursen, T.A., Lindberg, M., Taylor, R.L., 1999, "Vehicle dynamics simulations with coupled multi-body and finite element models", Finite Elements in Analysis and Design 31, pp 295-315.

[5] Bae DS and Haug EJ., 1987, "A recursive formulation for constrained mechanical system dynamics: Part 1. Open Loop System", Mechanics of Structure and Machines, 15(3):359-382.

[6] Kim SS and Haug EJ., 1988, "A recursive formulation for flexible multibody dynamics. Part I: Open loop system", Computer Methods in Applied Mechanics and Engineering, 71:481-506.

[7] Bae DS, Han JM and Yoo HH., 1999, "A generalized recursive formulation for constrained mechanical system dynamic", Mechanics of Structure and Machine, 27(3): 293–315

[8] Slaats PMA., 1991, "Recursive Formulations in Multibody Dynamics", Computational Mechanics.

[9] Rahnejat H., 1998, *Multi-body dynamics: vehicles, machines and mechanism*. Professional Engineering Publishing.

[10] Avello A, Jimenez JM, Bayo E and Garcia de Jalon J., 1993, “A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformation”, Computer Methods in Applied Mechanics and Engineering 107. North Holland, p. 313-339.

[11] Jimenez J.M., Avello A.N., Garcia De Jalon J. and Avello A.L., 1995, “An efficient implementation of the velocity transformation method for real-time dynamics with illustrative examples”, Computational Dynamics in Multibody Systems, pp. 15-35.

[12] Watton J., 1989, *Fluid Power Systems*. Prentice Hall International (UK) Ltd.

[13] Handroos HM and Vilenius MJ, 1991, “Flexible Semi-Empirical Models for Hydraulic Flow Control Valves”, Journal of Mechanical Design, 113(3):232–238.

[14] Baharudin, E.B., Rouvinen, A., Korkealaakso, P., and Mikkola A., 2014, “Real-time Multibody Application for Tree Harvester Truck Simulator”, Journal of Multi-body Dynamics, 11(100), pp. 67-85.

[15] Canudas de Wit, C., Tsiotras, P., Velenis, E., Basset, M., Gissinger G., 2003, “Dynamics friction models for longitudinal road/tire interaction”, Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 39(3), pp. 189-226.

[16] Do, N., Ferri, A., Bauchau O., 2007, “Efficient simulation of a dynamic system with LuGre friction”, Journal of Computational and Nonlinear Dynamics, 2(4), pp. 281-289.

[17] Olsson H., Astrom K.J., Canudas de Wit C., Gafvert M., Lischinsky P., 1998, “Friction models and friction compensation”, European Journal of Control, 4, pp. 176-195.

[18] F. Millo, L. Rolando, and M. Andreata, 2011, “Numerical simulation for vehicle powertrain development, numerical analysis - theory and application,” Prof. Jan Awrejcewicz (Ed.), InTech, Sep. Online: http://www.intechopen.com/books/numericalanalysis-theory-and-application/numerical-simulation-for-vehicle-powertraindevelopment.

[19] S. Zoroofi, 2008, *Modeling and simulation of vehicular power systems*, Master’s thesis, Chalmers University of Tehnology, Göteborg, Sweden.

[20] C. Mayet, L. Horrein, A. Bouscayrol, P. Delarue, J.-N. Verhille, E. Chattot, B. Lemaire-Semail, 2014, “Comparison of Different Models and Simulation Approaches for the Energetic Study of a Subway”, IEEE Transactions on Vehicular Technology, Feb., vol. 63, no. 2, pp.556–565.

[21] V. Ruuskanen, P. Immonen, J. Nerg, and J. Pyrhönen, 2012, “Determining electrical efficiency of permanent magnet synchronous machines with different control methods,” Electrical Engineering (Archiv fur Elektrotechnik), Jun., vol. 94, no. 2, pp. 97–106.

[22] M. Hafner, T. Finken, M. Felden, K. Hameyer, 2011, "Automated Virtual Prototyping of Permanent Magnet Synchronous Machines for HEVs," IEEE Transactions on Magnetics, May, vol. 47, no. 5, pp. 1018–1021.

[23] P. Vas, 1998. *Sensorless Vector and Direct Torque Control*. Oxford University Press, USA.

[24] M. Alaküla, 2006, *Hybrid Dirve system for Vehicles Part I*, Lund Tekniska Högskola.

[25] A. Bouscayrol, R. Schoenfeld, G. Dauphin-Tanguy, G.-H. Geitner, X. Guillaud, A. Pennamen, and J. Hautier, 2005, “Different energetic descriptions for electromechanical systems,” in Proc. of the European Conference on Power Electronics and Applications (EPE), Dresden, Germany, Sep.

[26] K. Chen, A. Bouscayrol, and W. Lhomme, 2008, “Energetic macroscopic representation and inversion-based control: Application to an electric vehicle with an electrical differential,” Journal of Asian Electric Vehicles, Jun., vol. 6, no. 1, pp. 1097–1102.

[27] M. Lindgren, 2005, “A Transient Fuel Consumption Model for Non-road Mobile Machinery”, Biosystems Engineering, Jul., vol. 91 Is. 2, pp. 139-174.

[28] M. Lindgren, P.-A, Hansson., 2004,” Effects of Transient Conditions on Exhaust Emissions from two Non-road Diesel Engines”, 2003, Biosystems Engineering, Jan., vol. 87 Is. 1, pp. 57-66.

Publication V

Baharudin, E., Rouvinen, A., Korkealaakso, P., Matikainen, M., and Mikkola, A.
Real-time Analysis of Mobile Machines Using Sparse Matrix Technique

Reprinted with permission from
Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics
Accepted: 3 February 2016

Real-time analysis of mobile machines using sparse matrix technique

Ezral Baharudin[⋆], **Asko Rouvinen**[†], **Pasi Korkealaakso**[†],
Marko K. Matikainen[⋆], **Aki Mikkola**[⋆]

[⋆] Lappeenranta University of Technology
Skinnarilankatu 34, 53850 Lappeenranta, Finland
e-mails: ezral.baharudin@lut.fi, marko.matikainen@lut.fi, aki.mikkola@lut.fi

[†] Mevea Ltd. Laserkatu 6, 53850 Lappeenranta, Finland
e-mails: asko.rouvinen@mevea.com, pasi.korkealaakso@mevea.com

ABSTRACT

The use of modern multibody simulation techniques enables the description of complex products, such as mobile machinery, with a high level of detail while still solving the equations of motion in real time. Using the appropriate modelling and implementation techniques, the accuracy of real-time simulation can be improved considerably. Conventionally, in multibody system dynamics, equations of motion are implemented using the full matrices approach that does not consider the sparsity feature of matrices. With this implementation approach, numerical efficiency decreases when sparsity increases. In this study, a numerical procedure based on semi-recursive and augmented Lagrangian methods for real-time dynamic simulation is introduced. To enhance computing efficiency, an equation of motion is implemented by employing the sparse matrix technique.

Keywords: Multibody system dynamics, real-time simulation, sparse matrix technique, semi-recursive method, augmented Lagrangian method

1 Introduction

The computer simulation of dynamic systems has proved to be an effective tool that is being implemented increasingly in machine development. Knowing how the dynamic behaviour of a machine is affected by variations in the design variables is important and can readily be studied with a good computer simulation model [1, 2]. To assess the performance of a machine using computerised methods, the system dynamics must be solved. A number of studies on how to solve the dynamics of a machine can be found in the literature [3, 4]. The dynamic analysis could provide the estimate external forces affected to the position of the components in the multibody system and also the estimate external forces caused by the interaction of the multibody system with the environment such as contact forces and friction forces [5, 6]. Increased computing capacity makes it possible to solve models that define large and complicated systems incorporating realistic mechanical properties, such as joint clearance and large deformation [7].

When considering the dynamic performance of mobile machines, it is important to note that the experience level of operators plays a critical role. However, most simulation research has focused on the development of modelling methods for machine components, while studies that account for the influence of the operator have been given little attention. Generally, this is because there are no mathematical expressions to describe operator behaviour.

Realistic operator behaviour can be taken into account by employing sophisticated real-time simulation models. In real-time simulations, the operator is actively engaged in the dynamic performance of the machinery. A real-time simulator must feel and perform like a real machine to the operator. This can be achieved only if the real-time simulation model is accurate and couples the physics from all the relevant engineering disciplines.

In the real-time analysis, knowing specifically what type of matrices to deal with can give an advantage in writing the optimal program code for numerical analysis. Conventionally, the system has been modelled with the full matrices approach that does not consider the sparsity of matrices. Increasing the number of bodies in the system will increase the size of matrices.

In this study, the numerical procedure—based on semi-recursive and augmented Lagrangian methods for the real-time dynamic simulation of multi-rigid body system—is applied to dynamic analysis of mobile machines. The equations of motion can be described in the form of differential-algebraic equations (DAEs). This is a case when the kinematic constraints are accounted for by the Lagrange multipliers, for example. In this study, the equations of motion are presented directly in ordinary differential equations (ODEs) form using the penalty and augmented Lagrangian methods. These methods make possible to find solution of the equations of motion using ODE solver without commonly used index reduction for the converting DAEs into ODEs. The objective of this paper is to demonstrate that the sophisticated usage of the sparse matrix computing approach can improve the computational efficiency of real-time dynamic simulation. To this end, computer codes based on the semi-recursive and augmented Lagrangian methods were written using C programming with the implementation formats of both full and sparse matrices. The performances with both full and sparse matrices are compared against each other.

2 The semi-recursive method

In the semi-recursive method, kinematic properties—such as position, velocity, and the acceleration of body particles—are developed based on the relative coordinates between neighbouring bodies connected by a joint [8]. This algorithm has been used and extended by several researchers and has been generalised to improve its implementation and efficiency [9, 10]. In this study, the velocity transformation matrix is used in developing the equations of motion. This method used a minimal set coordinates instead of full generalized coordinates which could reduce the size of system matrices in equations of motion.

2.1 Kinematics

Relative motion between both neighbouring bodies and constraints are the two main aspects of recursive kinematics that are used to generate the equations of motion for a multibody system. The method uses the global position and local rotational coordinates of the centre of gravity as the generalised coordinates used in the formulation of the equations of motion. The multibody system of two bodies interconnected by a joint is illustrated in Figure 1.

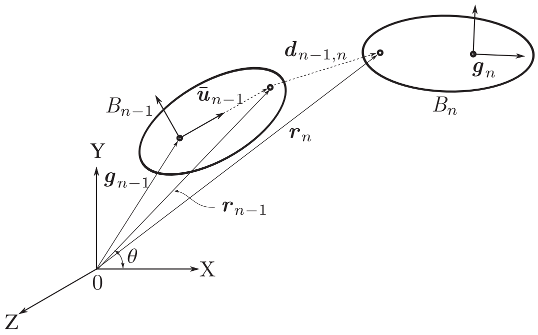


Figure 1. A description of the kinematics for the recursive method.

The position of joint $\boldsymbol{r}_n$ for the body B_n can be described using kinematic of the previous body B_{n-1} in the global frame as follows,

$$\boldsymbol{r}_n = \boldsymbol{g}_{n-1} + \boldsymbol{A}_{n-1}\bar{\boldsymbol{u}}_{n-1} + \boldsymbol{d}_{n-1,n} \tag{1}$$

where $\boldsymbol{g}_{n-1}$ is the position vector of the centre of gravity of body B_{n-1}, $\boldsymbol{A}_{n-1}$ is the rotation matrix of the body B_{n-1}, $\bar{\boldsymbol{u}}_{n-1}$ is the constant position vector within the local co-ordinate system, and $\boldsymbol{d}_{n-1,n}$ is relative displacement vector between bodies. The velocity vector of joint $\dot{\boldsymbol{r}}_n$ for the body B_n can be determined as follows:

$$\dot{\boldsymbol{r}}_n = \dot{\boldsymbol{g}}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1}\bar{\boldsymbol{u}}_{n-1} + \dot{\boldsymbol{d}}_{n-1,n}, \tag{2}$$

The rotation matrix and the skew-symmetric matrix of the angular velocity vector for body B_n can be computed as

$$\boldsymbol{A}_n = \boldsymbol{A}_{n-1}\boldsymbol{A}_{n-1,n} \tag{3}$$

$$\tilde{\boldsymbol{\omega}}_n = \tilde{\boldsymbol{\omega}}_{n-1} + \tilde{\boldsymbol{\omega}}_{n-1,n} \tag{4}$$

where $\boldsymbol{A}_{n-1,n}$ and $\tilde{\boldsymbol{\omega}}_{n-1,n}$ are the relative rotation matrix and the skew-symmetric matrix of the relative angular velocity vector.

2.2 The equations of motion

The equations of motion for the body n can be presented with respect to the centre of gravity using the Newton-Euler approach, as follows,

$$\underbrace{\begin{pmatrix} m_n\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_n \end{pmatrix}}_{\boldsymbol{M}_n} \underbrace{\begin{pmatrix} \ddot{\boldsymbol{g}}_n \\ \dot{\boldsymbol{\omega}}_n \end{pmatrix}}_{\ddot{\boldsymbol{q}}_n} + \underbrace{\begin{pmatrix} \mathbf{0} \\ \tilde{\boldsymbol{\omega}}_n \boldsymbol{J}_n \boldsymbol{\omega}_n \end{pmatrix}}_{\boldsymbol{Q}_n^v} = \underbrace{\begin{pmatrix} \boldsymbol{F}_n \\ \boldsymbol{T}_n \end{pmatrix}}_{\boldsymbol{Q}_n^e}. \tag{5}$$

where $\mathbf{I}$ is an identity matrix, $\boldsymbol{M}_n$ is the mass matrix, $\ddot{\boldsymbol{q}}_n$ is the vector of translational and angular accelerations, $\boldsymbol{Q}_n^v$ accounts for the centrifugal terms and $\boldsymbol{Q}_n^e$ is the vector of the external forces $\boldsymbol{F}_n$ and torques $\boldsymbol{T}_n$. The body's properties with respect to the inertia, and centrifugal and external forces of the system of n bodies can be written as:

$$\boldsymbol{M} = \mathrm{diag}(\boldsymbol{M}_1, \boldsymbol{M}_2, \dots, \boldsymbol{M}_n) \tag{6}$$

$$\ddot{\boldsymbol{q}} = [\ddot{\boldsymbol{q}}_1{}^T, \ddot{\boldsymbol{q}}_2{}^T, \dots, \ddot{\boldsymbol{q}}_n{}^T]^T \tag{7}$$

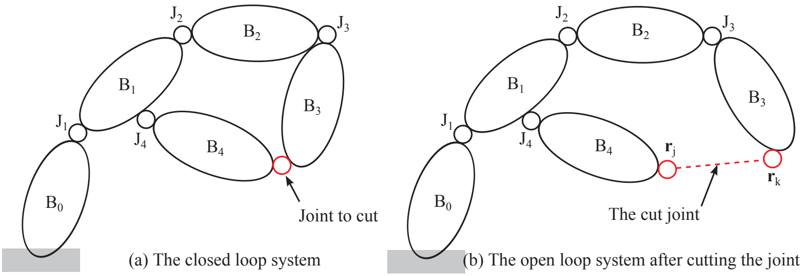
$$\boldsymbol{Q}^v = [\boldsymbol{Q}_1^{vT}, \boldsymbol{Q}_2^{vT}, \dots, \boldsymbol{Q}_n^{v\,T}]^T \tag{8}$$

$$\boldsymbol{Q}^e = [\boldsymbol{Q}_1^{eT}, \boldsymbol{Q}_2^{eT}, \dots, \boldsymbol{Q}_n^{e\,T}]^T \tag{9}$$

When velocity transformation, $\boldsymbol{R}$ in case of the scleronomic system $\dot{\boldsymbol{q}} = \boldsymbol{R}\dot{\boldsymbol{z}}$ is taken account and Newton-Euler equations are multiplied by $\boldsymbol{R}^T$, the equations of motion can be presented as follows,

$$\underbrace{\boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}}_{\boldsymbol{M}^*}\ddot{\boldsymbol{z}} = \underbrace{\boldsymbol{R}^T(\boldsymbol{Q}^e - \boldsymbol{M}\boldsymbol{c} - \boldsymbol{Q}^v)}_{\boldsymbol{Q}^*} \tag{10}$$

where in the case of scleronomic constraints $\boldsymbol{c} = \dot{\boldsymbol{R}}\dot{\boldsymbol{z}}$, and $\boldsymbol{M}^*$ and $\boldsymbol{Q}^*$ are the mass matrix and force vectors respectively, represented with respect to the relative coordinates. From the literature [11], the matrix $\dot{\boldsymbol{R}}$ does not need to be calculated explicitly. In fact, it is not computationally efficient to define it explicitly. Therefore, the coordinates $\ddot{\boldsymbol{z}}$ for generalised acceleration vectors, $\ddot{\boldsymbol{q}} = \boldsymbol{R}\ddot{\boldsymbol{z}} + \dot{\boldsymbol{R}}\dot{\boldsymbol{z}}$ should be set to zero, which results in the relationship $\dot{\boldsymbol{R}}\dot{\boldsymbol{z}} = \ddot{\boldsymbol{q}}$.



(a) The closed loop system

(b) The open loop system after cutting the joint

Figure 2. The closed loop system and the cut joint.

When the multibody system includes closed loops as in Figure 2(a), the system need to be treated as an open loop system by cutting the selected joint, as shown in Figure 2(b).

After that, the cut joint between two nodes, r_j and r_k are modelled using the constraints. When the constraints $\boldsymbol{\Phi}$ (due to a closed loop) are accounted for with the penalty method, the equations of motion as a function of the relative coordinates $\boldsymbol{z}$ can take form for the semi-recursive method as follows,

$$\boldsymbol{M}^*\ddot{\boldsymbol{z}} = \boldsymbol{Q}^* - \boldsymbol{\alpha}\boldsymbol{\Phi}_{,z}^T(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\Omega}\boldsymbol{\mu}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}^2\boldsymbol{\Phi}) \tag{11}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}$ and $\boldsymbol{\mu}$ are diagonal matrices including the penalty factors, $\boldsymbol{\Phi} = \boldsymbol{\Phi}(\boldsymbol{z}(t))$ is the vector of constraint equations, and $\boldsymbol{\Phi}_{,z}$ is the Jacobian matrix of the constraint equations with respect to the relative coordinates.

2.3 Velocity transformation

The bodies in a multibody system are often connected via components of the revolute and prismatic joints, and/or their combinations. As shown in Figure 2, each body of the system is labelled based on tree-structure multibody numbering. The ground body is labelled as B_0 (called as the base). The next bodies from the base to the end are numbered in increasing order, in the order in which they should meet $B_n > B_{n-1}$. This numbering is also applied to the labelling of the joints. In this multibody system, each joint can be represented in transformation velocity matrix notation $\boldsymbol{R}$. The formulation of joint variable $\boldsymbol{R}_n$ depends on the types of joints and the degrees of freedom (DOFs). The equations for revolute and prismatic joints proposed by Avello et al. [11] can be presented as

$$\boldsymbol{R}_n^{rev} = \begin{bmatrix} \tilde{\boldsymbol{e_n}}(\boldsymbol{g}_{B_n} - \boldsymbol{r}_i) \\ \boldsymbol{e}_n \end{bmatrix}; \quad \boldsymbol{R}_n^{pris} = \begin{bmatrix} \boldsymbol{e}_n \\ \boldsymbol{0} \end{bmatrix} \tag{12}$$

where $\boldsymbol{e}_n$ is the unit vector of joint n that point the direction of the revolute axis , $\tilde{\boldsymbol{e}}_n$ is the skew-symmetric matrix of $\boldsymbol{e}_n$ and $(\boldsymbol{g}_{B_n} - \boldsymbol{r}_i)$ are the vector points from point i in body n to the centre of gravity of designated body B_n.

To get the transformation velocity matrix $\boldsymbol{R}$ for the assembled system, all dedicated matrix $\boldsymbol{R}_n$ need to be arranged into the system level matrix that represents all joints. The rows of velocity matrix $\boldsymbol{R}$ represent the related body and the columns of velocity matrix $\boldsymbol{R}$ represent the number of DOF found in the path from body j to the ground. Therefore, for the example model shown in Figure 2(b) with an assumption all joints are revolute joint, the transformation velocity matrix $\boldsymbol{R}$ can be arranged as

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_1 & & & 0 \\ \boldsymbol{R}_1 & \boldsymbol{R}_2 & & \\ \boldsymbol{R}_1 & \boldsymbol{R}_2 & \boldsymbol{R}_3 & \\ \boldsymbol{R}_1 & 0 & 0 & \boldsymbol{R}_4 \end{bmatrix} \tag{13}$$

where $\boldsymbol{R} \in \mathbb{R}^{24\times6}$. As an example, to see the dedicated matrix $\boldsymbol{R}$ corresponding to the joint at body B_3, it can be extracted from above equation as

$$\boldsymbol{R}_{B_3} = \left[\begin{array}{ccc} \boldsymbol{R}_1 & \boldsymbol{R}_2 & \boldsymbol{R}_3 \end{array} \right] \tag{14}$$

where $\boldsymbol{R}_{B_3} \in \mathbb{R}^{6\times3}$. The mass matrix of the system $\boldsymbol{M} = \mathrm{diag}(\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_4)$ where the equation for the mass matrix where $\boldsymbol{M}_n \in \mathbb{R}^{6\times6}$ for the body n is written as

$$\boldsymbol{M}_n = \left[\begin{array}{cc} m_n\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_n \end{array} \right] \tag{15}$$

where m_n and $\boldsymbol{J}_n$ describe mass and inertia properties of body n, respectively. It can be seen from the equations of motion in equation (10) that coordinate transformation is employed as $\boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}$ to reduce the dimensions of system matrices. To increase the computational efficiency, some operations can be solved in parallel, as is suggested by Jiménez et al. [12].

3 The augmented Lagrangian method

Penalty methods are problematic due to the difficulty of choosing good estimates for penalty factors. The penalty values should be large enough for fullfilling constraints within specified tolerances. Otherwise, choosing overly large penalty factors leads to an ill-conditioned problem that increases errors in the numerical solution. One possibility for solving the constrained optimisation problem is to use formulations based on the augmented Lagrangian methods, which can be traced back to the mid-1940s [13]. The augmented Lagrangian methods combine the advantages of the Lagrangian multiplier and penalty methods. In the augmented Lagrangian methods, the Lagrangian of the constrained problem is augmented with additional penalty terms. The use of the widely-used augmented Lagrangian method in real-time dynamic simulation of multibody systems is explained in details in [14]. The method can be derived by introducing variational terms due to fictitious energies and the Lagrange multipliers, as follows,

$$\begin{aligned} \delta W^*_{pot} &= (\dot{\boldsymbol{\Phi}}^T_{,q}\boldsymbol{\alpha}\dot{\boldsymbol{\Phi}} - \boldsymbol{\Omega}^2\boldsymbol{\Phi}^T_{,q}\boldsymbol{\alpha}\dot{\boldsymbol{\Phi}}) \cdot \delta\boldsymbol{q} \\ \delta W^*_{kin} &= \dot{\boldsymbol{\Phi}}^T_{,q}\boldsymbol{\alpha}\dot{\boldsymbol{\Phi}}^T_{,q} \cdot \delta\boldsymbol{q} \\ \delta W^*_{dis} &= -2\boldsymbol{\alpha}\boldsymbol{\Omega}\boldsymbol{\mu}\boldsymbol{\Phi}^T_{,q}\dot{\boldsymbol{\Phi}} \cdot \delta\boldsymbol{q} \\ \delta W_\lambda &= \lambda\boldsymbol{\Phi}^T_{,q} \cdot \delta\boldsymbol{q} \end{aligned} \tag{16}$$

where δW^*_{pot}, δW^*_{kin}, δW^*_{dis}, and δW_λ are the virtual work of potential energy, kinematic energy, dissipative energy and lagrange multipliers while $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}$ and $\boldsymbol{\mu}$ are diagonal matrices including the penalty factors, $\boldsymbol{\Phi} = \boldsymbol{\Phi}(\boldsymbol{q}(t))$ is the vector of constraint equations, and $\boldsymbol{\Phi}_{,q}$ is the Jacobian matrix of the constraint equations with respect to the generalised coordinates. When equations are taken into account in Hamilton′s principle, the equations of motion can be written in the form of index-3 as

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{Q} + \boldsymbol{\alpha}\boldsymbol{\Phi}_{,q}^{T}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\Omega}\boldsymbol{\mu}\dot{\boldsymbol{\Phi}} + \boldsymbol{\Omega}^{2}\boldsymbol{\Phi}) + \boldsymbol{\Phi}_{,q}^{T}\boldsymbol{\lambda} = \boldsymbol{0} \\ \boldsymbol{\Phi}(\boldsymbol{q}(t)) = \boldsymbol{0}. \end{cases} \tag{17}$$

where the force vector $\boldsymbol{Q} = \boldsymbol{Q}_e - \boldsymbol{Q}_v$. As can be seen from Equation (17), the penalty terms are corrected by the Lagrange multipliers so that new values for multipliers can be estimated iteratively, as follows,

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \boldsymbol{\alpha}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\Omega}\boldsymbol{\mu}\dot{\boldsymbol{\Phi}} + \boldsymbol{\Omega}^{2}\boldsymbol{\Phi})_{k+1}; \quad k = 0, 1, 2, \ldots \tag{18}$$

where k is the number of the iteration step and the initial quess $\boldsymbol{\lambda}_0 = 0$ for the first iteration. For this reason, the penalty factors do not need to be large. Due to the iterative solution of the Lagrange multipliers, the augmented Lagrangian formulation leads to a system of ordinary differential equations without additional unknowns. The system of the equations of motion, including an iterative scheme, can be written in the form of index-1 as follows

$$(\boldsymbol{M} + \boldsymbol{\Phi}_{,q}^{T}\boldsymbol{\alpha}\boldsymbol{\Phi}_{,q})\ddot{\boldsymbol{q}}_{k+1} = \boldsymbol{M}\ddot{\boldsymbol{q}}_k - \boldsymbol{\alpha}\boldsymbol{\Phi}_{,q}^{T}(\dot{\boldsymbol{\Phi}}_{,q}\dot{\boldsymbol{q}} + 2\boldsymbol{\Omega}\boldsymbol{\mu}\dot{\boldsymbol{\Phi}} + \boldsymbol{\Omega}^{2}\boldsymbol{\Phi}) \tag{19}$$

where $\boldsymbol{M}\ddot{\boldsymbol{q}}_0 = \boldsymbol{Q}$ for the first iteration. More details about the augmented Lagrangian formulations for imposing constraints in multibody systems can be found in [14, 15]. However, this iterative process needs to meet the tolarance value in order to control the iteration period:

$$\|\ddot{\mathrm{q}}_{k+1} - \ddot{\mathrm{q}}_k\| = \varepsilon \tag{20}$$

where it can be solved using Newton-Raphson iteration.

4 The sparse matrix and solvers

In the equations of motion shown previously, some of the matrices involved have a tendency to be sparse in structure. In this equation, there are two matrices that potentially have sparse properties: the mass matrix, $\boldsymbol{M}$, the Jacobian matrix of constraints $\boldsymbol{\Phi}_{,q}$ and the velocity transformation matrix $\boldsymbol{R}$.

The dimension of matrix $\boldsymbol{M}$ becomes more radical for a spatial model as it will increase in dimension compared to a planar model. The sparsity of the mass matrix can be described as shown in Figure 3. The number of zero entry increases exponentially with respect to the number of bodies. In the case under consideration, the mass matrix is symmetric and positive definite.

The sparsity pattern also can be found in the Jacobian matrix of constraints $\boldsymbol{\Phi}_{,q} \in \mathbb{R}^{n_c \times n_g}$ where n_c is the number of constraint equations, and n_g is the number of generalised coordinates. For example, if the multibody system have a multiple joints in the spatial system,

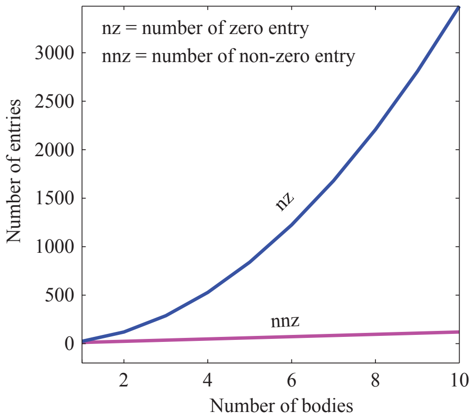


Figure 3. Sparsity description for the mass matrix, $\boldsymbol{M}$.

constraint equations of all joints need to be derived. The constraint equation of joint is formulated based on the two bodies connected at the joint. If the joint is a revolute joint, five constraint equations need to be added into the vector of constraint $\boldsymbol{\Phi}$. If the system has n_r number of revolute joints, the number of constraint equations increases to $5n_r$. Therefore, it can be said the Jacobian matrix $\boldsymbol{\Phi}_{,q}$ in a large scale constrained system will become sparse and not symmetric.

In the case of the velocity transformation matrix, the sparsity of the velocity transformation matrix $\boldsymbol{R}$ depends on the type of joint, number of joints and the arrangement of the bodies. If the topology structure is a tree structure, the sparsity increases with the number of branches. As the assemble matrix $\boldsymbol{R}$ is based on the topology structure, the matrix $\boldsymbol{R}$ can be lower or upper triangular as can be seen in equation (13).

Solving a sparse linear system, $\boldsymbol{A}_{spd}\boldsymbol{x} = \boldsymbol{b}$ where matrix $\boldsymbol{A}_{spd}$ is symmetric positive-definite can be done using direct or iterative methods. The direct method will factorise the sparse matrix and use a reordering scheme to solve the system. The iterative method uses an approximation solution through residual to obtain the final solution. In this study, the direct method is chosen as a method to solve a linear system.

The most common solver in the direct method which is Cholesky decomposition is used in this study. In this approach, a symmetric positive-definite of matrix $\boldsymbol{A}_{spd}$ is factorised into the product of a lower triangular matrix and its transpose, where $\boldsymbol{A}_{spd} = \boldsymbol{C}\boldsymbol{C}^T$ and $\boldsymbol{C}$ is called the Cholesky factor [16]. Matrix $\boldsymbol{A}_{spd}$ is transformed into the Cholesky factor $\boldsymbol{C}$ using two equations as:

$$C_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} C_{kj}^2} \tag{21}$$

$$C_{ik} = \frac{1}{C_{kk}}\left(a_{ik} - \sum_{j=1}^{k-1} C_{ij}C_{kj}\right) \tag{22}$$

where C_{kk} is the diagonal elements of matrix $\boldsymbol{C}$, a_{kk} is the diagonal element of matrix $\boldsymbol{A}_{spd}$, and C_{ik} is the elements below the diagonal of matrix $\boldsymbol{C}$ where $i > k$. From here, the triangular system can be used to solve $\boldsymbol{Cy} = \boldsymbol{b}$ for $\boldsymbol{y}$ and then finally solve $\boldsymbol{C}^T\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$.

Programming

Therefore, it is beneficial to solve the equations of motion using the sparse matrix technique. Applying a sparse implementation approach, data storage management, and a sparse procedure can increase the computing efficiency and optimise memory storage [17].

The main idea of the sparse technique is to reduce memory space by storing the non-zero elements in contiguous memory locations for more efficient execution. Therefore, the type of sparse format plays an important role in terms of computing efficiency. Due to the solver being written in C code, there are several sparse formats that could be used while the optimal format selection depends on the matrix structure. In this study, the matrices have been stored using the compressed sparse column (CSC) format, which is commonly and widely used in sparse solver UMFPACK and ARPACK library packages for example. In this format, three arrays—which are the element ($Elem$), row indices (Row_Ind), and the column pointer (Col_Ptr) —are specified. The array of $Elem$ $[1, \ldots, k]$ stores the non-zero elements of the matrix, Row_Ind $[1, \ldots, k]$ stores the row indices of each non zero element, and Col_Ptr $[1, \ldots, n]$ stores the index of the element in $Elem$ which starts a column of the matrix. Figure 4 shows an example of CSC format.

Full matrix

$$\begin{bmatrix} 2 & 0 & 0 & 10 & 0 & 12 \\ 0 & 2 & 0 & 0 & 6 & 2 \\ 0 & 0 & 55 & 77 & 0 & 98 \\ 23 & 0 & 21 & 32 & 0 & 2 \end{bmatrix}$$

Compressed Sparse Column

Col_Ptr	1		3	4		6			9				10	14
Row_Ind	1	4	2	3	4	1	3	4	2	1	2	3	4	
Elem	2	23	2	55	21	10	77	32	6	12	2	98	2	

Figure 4. Converting a full matrix into CSC format.

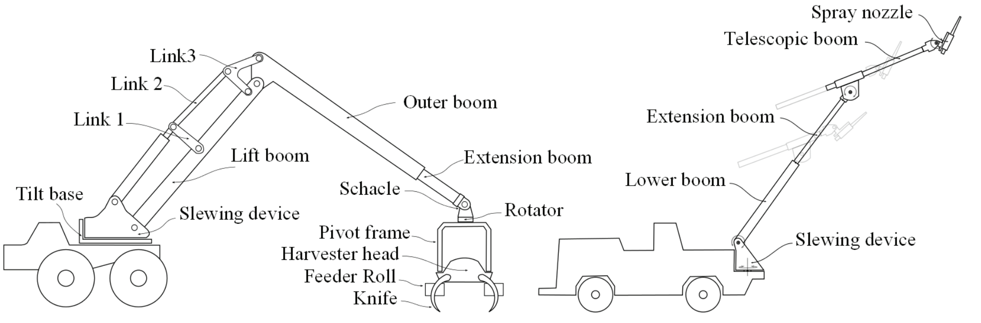
A reliable and robust libraries, UMFPACK (using the unsymmetrical multifrontal method to solve the linear system), and a CSPARSE package can be used for as a direct solver for linear system. CSPARSE is used for the basic matrix operations [18], matrix structure manipulation and matrix storage format, while UMFPACK is used for solving the form of Ax = b using the unsymmetrical mutifrontal method [19]. ARPACK is a numerical library for solving large eigenvalue problems using the implicity restarted Arnoldi method, which is fast and robust solver [20]. ARPACK also needs two other libraries in order to work;

Blas and SuperLU. In this case, ARPACK is an optional solver if the eigen analysis need to be done.

5 Numerical examples and discussions

The numerical problems introduced in this study are carried out using the Mevea solver. The solver is coded to treat the matrices with full or sparse formats. To determine the efficiency of augmented Lagrangian formulation and semi-recursive methods when using a sparse matrix approach, the mobile working machine the Logset H8 tree harvester and the Normet Concrete Spray System (NCSS) are analysed. The time integration for the equations of motion is obtained by using the explicit Runge-Kutta method of the fourth order with a fixed time step of 0.8 for the NCSS and 1.6 milliseconds for the Logset H8.

Figure 5 show the depiction of multibody systems of the Logset H8 tree harvester and the Normet Concrete Spray System which are used to compared the computing performance when using different methods. Figure 6 show an example of the topology structure of the Logset H8 tree harvester where the system is treated as an open loop system by an implemented cut-joint approach in the closed loop chain and the cut-joint nodes are imposed with a constraint using the penalty method.



(a) Logset H8 Tree Harvester

(b) Normet Concrete Spray System

Figure 5. A multibody depiction of (a) the Logset H8 Tree Harvester and (b) the Normet Concrete Spray System.

The sparsity of the matrices involved in the analysis can be seen from Table 1. The number of bodies for the NCSS and Logset H8 are 11 and 30 correspondingly, and they have 10 and 27 DOFs respectively. The sparsity percentage of the number of elements in the Jacobian of constraints and mass matrices decreases when the DOFs increase. The transformation velocity matrix for the system $\boldsymbol{R}$ is only used in semi-recursive method, where the sparsity percentages are 8.33% for the NCSS, and 7.14% for Logset H8.

The computational efficiency for the augmented Lagrangian and semi-recursive methods with and without the sparse matrix technique are illustrated in Figure 7 and Figure 8 for the NCSS and Logset H8 correspondingly. In order to see the performance of each

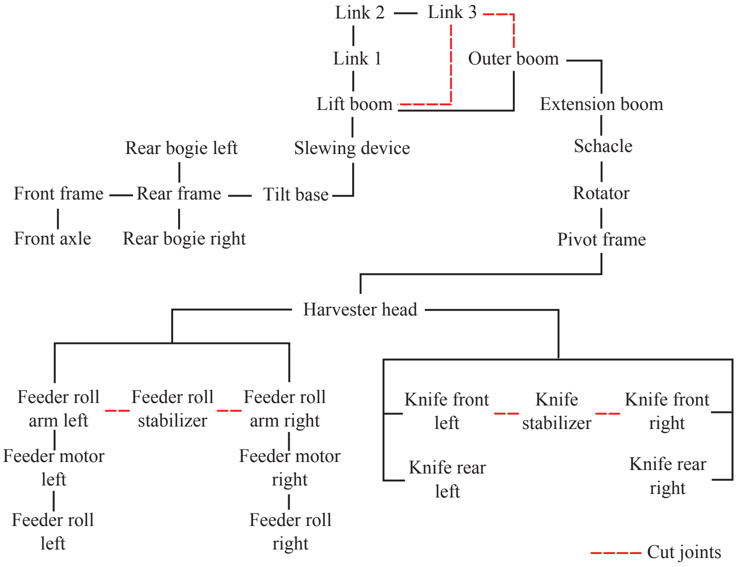


Figure 6. A detailed topology of the Logset H8 tree harvester.

method with and without the sparse technique, the computation loop duration is recorded for every solution and synchronised with the real clock time. Both figures shows the loop duration per step to solve the equations of motion for 20 seconds and are compared with the fixed time step of the fourth order Runge-Kutta method. For the NCSS, the sparse technique enhanched the computing efficiency for both augmented Lagrangian and semi-recursive methods. As can be seen, the semi-recursive method in full matrices approach is computationally more efficient when compared to the augmented Lagrangian method. This is due to the small number of DOFs, which lead to the small size of matrices. For the Logset H8, the augmented Lagrangian method needed additional time to solve the linear system at each loop but when the system matrices are implemented with the sparse technique, the computational time did not shows any significant improvement. For the NCSS solution, when the semi-recursive method is applied, the computational time at each loop reduced significantly when compared with the augmented Lagrangian method. However, when the matrices are executed with the sparse technique, the computional time at each loop slightly increases but still less than the reference time step.

Figure 9 shows the average loop duration for two methods, both with and without the sparse technique, and compares them againts the fixed time step. In the real-time simulation, the simulation time must be synchronised with the real clock time at the end of each time segment. For the NCSS model, the usage of the sparse implementation technique increases the computational time for both the augmented Lagrangian method and the semi-recursive method, and it reduces the computing efficiency to 42.52% and 9.16%, correspondingly when compared to full matrix implementation. For the Logset H8 model, the sparse implementation technique in the augmented Lagrange method helps to reduce the computing time by 5.75% when compared to full matrix implementation. In the case of the semi-recursive method, the sparse implementation technique reduces the computing

Table 1. Sparsity description data.

Description	Normet concrete spray		H8 Tree harvester	
	Lagrangian	Recursive	Lagrangian	Recursive
Number of the bodies	11	11	30	30
generalised coordinates	66	66	180	180
Joint coordinates	-	12	-	35
Constraint equations	56	2	153	8
Degree of freedoms (DOFs)	10	10	27	27
Elements in the Jacobian (eJac)	3696	132	27540	1440
Non-zero elements (nzeJac)	432	24	1128	78
nzeJac/eJac (%)	11.69	18.18	4.1	5.42
Elements in $\boldsymbol{M}$ (eM)	4356	4356	32400	32400
Non-zero elements in $\boldsymbol{M}$ (nzeM)	330	330	900	900
nzeM/eM (%)	7.58	7.58	2.78	2.78
Elements in $\boldsymbol{R}$ (eR)	-	792	-	6300
Non-zero elements in $\boldsymbol{R}$ (nzeR)	-	66	-	450
nzeR/nR (%)	-	8.33	-	7.14

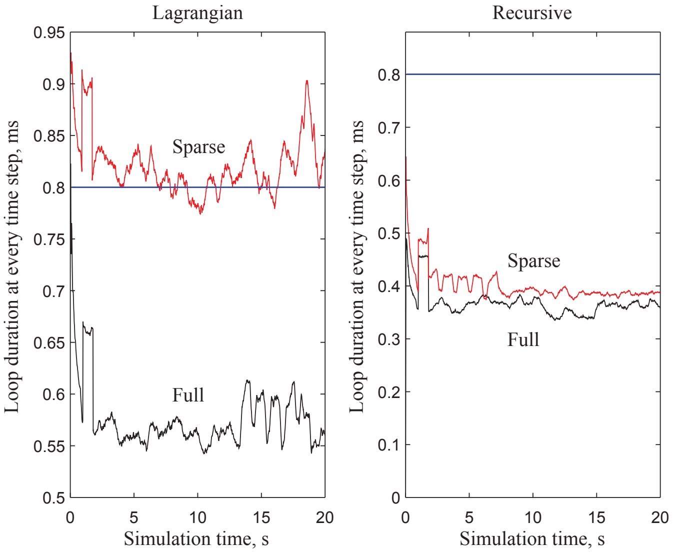


Figure 7. Loop durations at the time step 0.8ms of the augmented Lagrangian method vs. the semi-recursive method of the NCSS, solved with both full and sparse matrix techniques. Continuous straight lines illustrate a constant step size of the fourth order Runge-Kutta method with respect to loop durations.

time efficiency by 4.55% as compared to full matrix implementation

It can be concluded from the numerical examples that augmented Lagrangian method, when compared to semi-recursive method, requires additional time to solve the equations

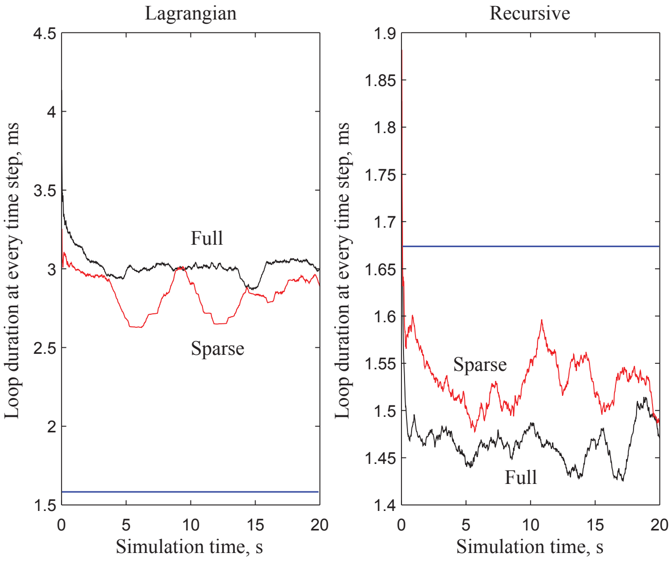
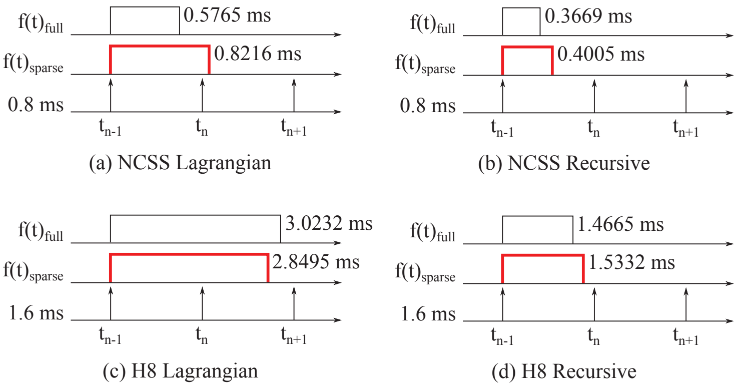


Figure 8. Loop durations at the time step 1.6ms of the augmented Lagrangian method vs. the semi-recursive method of the Logset H8 tree harvester. Continuous straight lines illustrate a constant step size of the fourth order Runge-Kutta method with respect to loop durations.



(a) NCSS Lagrangian

(b) NCSS Recursive

(c) H8 Lagrangian

(d) H8 Recursive

Figure 9. The average loop duration for each method compared to the fixed time step.

of motion when the number of DOFs and number of entries in the matrices′ system are large. The semi-recursive method seems to be less able to solve the equations of motion compared to the fixed time step but the implementation of the sparse technique may not help significantly in improving efficiency. However, when both methods (with or without sparse implementation) for both system models are compared side by side, the semi-recursive method is about 50% more efficient than the augmented Lagrangian method in the studied cases.

6 Summary and conclusion

The real-time simulation solver for the dynamics analysis of multibody systems has been introduced based on two methods; the augmented Lagrangian method and the semi-recursive method. The system matrices of both methods are implemented with and without the sparse technique in order to see the performance changes. The solver is written in C programming language with help of reliable and robust library packages to ensure the outcomes are accurate.

Numerical examples shows that the implementation of the semi-recursive method with a transformation velocity matrix produced a significant improvement to the computing efficiency when compared to the augmented Lagrangian method in the studied cases. When applying the sparse matrix technique in the two methods, the computing efficiency improved for certain methods. It was also noticed that in some applications the usage of the sparse implementation technique even decreased computational efficiency. Therefore, it should be noted that the computational efficiency when using the sparse technique is strongly case dependent. As is known, larger-sized system matrices have increased sparsity and, therefore, more computational benefit can be gained from the use of thfae sparse matrix technique in multibody applications. Further investigation into sparse implementation is needed in order to maximise efficiency.

Acknowledgement

The authors would like to thank the Academy of Finland (Application No. 259543 for the funding of Postdoctoral Researchers) for supporting Marko K. Matikainen.

REFERENCES

[1] Moon FC. Applied dynamics: with applications to multibody and mechatronic systems. John Wiley & Sons; 2008.

[2] Schiehlen W, Guse N, Seifried R. Multibody dynamics in computational mechanics and engineering applications. Computer Methods in Applied Mechanics and Engineering. 2006;195(41):5509–5522.

[3] Schiehlen W. Multibody system dynamics: roots and perspectives. Multibody system dynamics. 1997;1(2):149–188.

[4] Rahnejat H. Multi-body dynamics: vehicles, machines, and mechanisms. London: Professional Engineering Publishing; 1998.

[5] Schiehlen WO. Multibody systems handbook. vol. 6. Berlin: Springer; 1990.

[6] Eich-Soellner E, Führer C. Numerical methods in multibody dynamics. vol. 45. Wiesbaden: Springer; 1998.

[7] Shabana AA. Flexible multibody dynamics: review of past and recent developments. Multibody system dynamics. 1997;1(2):189–222.

[8] Korkealaakso PM, Rouvinen AJ, Moisio SM, Peusaari JK. Development of a real-time simulation environment. Multibody System Dynamics. 2007;17(2–3):177–194.

[9] Bae DS, Haug EJ. A recursive formulation for constrained mechanical system dynamics: Part 1. open loop system. Mechanics of Structure and Machines. 1987;15(3):359–382.

[10] Bae DS, Han JM, Yoo HH. A generalized recursive formulation for constrained mechanical system dynamic. Mechanics of Structure and Machine. 1999;27(3):293–315.

[11] Avello A, Jimenez JM, Bayo E, de Jalón JG. A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformation. Computer Methods in Applied Mechanics and Engineering. 1993;107(3):313–339.

[12] Jiménez JM, Avello AN, de Jalón JG, Avello AL. An efficient implementation of the velocity transformation method for real-time dynamics with illustrative examples. In: Computational Dynamics in Multibody Systems. Netherland: Springer; 1995. p. 15–35.

[13] Gill PE, Murray W, Wright MH. Practical optimization. London: Academic Press Inc.; 1981.

[14] Bayo E, de Jalón JG. Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge. New York: Springer-Verlag; 1993.

[15] Bayo E, de Jalón JG, Serna MA. A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Computer Methods in Applied Mechanics and Engineering. 1988;71(2):183–195.

[16] Loan CFV. Introduction to scientific computing: A matrix-vector approach using MATLAB. New Jersey: Prentice Hall; 1999.

[17] Pissanetzky S. Sparse matrix technology. Orlando: Academic Press; 1984.

[18] Davis TA. Direct methods for sparse linear systems. vol. 2. Philadelphia, USA: Society for Industrial and Applied Science; 2006.

[19] Davis TA. UMFPACK version 5.2.0 user guide. Gainesville: University of Florida; 2007.

[20] Lehoucg RB, Sorensen DC, Yang C. ARPACK users' guide: Solution of large scale eigenvalue problems with Implicity Restarted Arnoldi Methods. Philadelphia: Society for Industrial and Applied Mathematics; 1998.

Nomenclatures

a_{kk}	diagonal elements of matrix $\boldsymbol{A}_{spd}$ at location (k,k)
a_{ik}	non-diagonal elements of matrix $\boldsymbol{A}_{spd}$ at location (i,k)
$\boldsymbol{A}_{n-1}$	rotation matrix of body n-1
$\boldsymbol{A}_{n-1,n}$	relative rotation matrix
$\boldsymbol{A}_{spd}$	symmetric positive definite matrix
B_n	body number n
B_{n-1}	body number n-1
$\boldsymbol{C}$	Cholesky factorisation matrix
C_{kk}	diagonal elements of matrix $\boldsymbol{C}$ at location (k,k)
C_{ik}	elements below diagonal elements of matrix $\boldsymbol{C}$ at location (i,k)
C_{ij}	elements below diagonal elements of matrix $\boldsymbol{C}$ at location (i,j)
C_{kj}	elements below diagonal elements of matrix $\boldsymbol{C}$ at location (k,j)
$\boldsymbol{d}_{n-1,n}$	relative displacement vector between bodies
$\dot{\boldsymbol{d}}_{n-1,n}$	relative velocity vector between bodies
$\boldsymbol{e}_n$	vector of joint n that point the direction of the joint axis
$\boldsymbol{F}_n$	external forces of body n
$\boldsymbol{g}_{n-1}$	position vector of the centre of gravity of body n-1
$\dot{\boldsymbol{g}}_{n-1}$	velocity vector of the centre of gravity of body n-1
$\ddot{\boldsymbol{g}}_n$	acceleration vector of the centre of gravity of body n-1
$\boldsymbol{g}_{B_n}$	position vector of the centre of gravity of considered body n
$\mathbf{I}$	identity matrix
$\boldsymbol{J}_n$	inertia of body n
k	iteration number
m_n	mass of body n
$\boldsymbol{M}_n$	mass matrix of body n
n_r	number of joints
$\dot{\boldsymbol{q}}$	vector of generalised velocities
$\ddot{\boldsymbol{q}}_n$	vector of generalised accelerations of body n
$\boldsymbol{Q}_n^v$	centrifugal forces of body n
$\boldsymbol{Q}_n^e$	applied forces of body n
$\boldsymbol{r}_n$	position vector of joint n
$\boldsymbol{r}_i$	position vector of point i
$\dot{\boldsymbol{r}}_n$	velocity vector of joint n
$\boldsymbol{R}$	velocity transformation matrix
$\dot{\boldsymbol{R}}$	first derivative of velocity transformation matrix
$\boldsymbol{R}_n^{rev}$	velocity transformation matrix for revolute joint n
$\boldsymbol{R}_n^{pris}$	velocity transformation matrix for prismatic joint n
$\boldsymbol{T}_n$	torques of body n
$\bar{\boldsymbol{u}}_{n-1}$	constant position vector within the local coordinate system
$\dot{\boldsymbol{z}}$	relative velocities of joint
$\ddot{\boldsymbol{z}}$	relative accelerations of joint

Greek symbols

$\boldsymbol{\alpha}$	penalty factor
$\boldsymbol{\Omega}$	penalty factor with respect to natural frequency
$\boldsymbol{\mu}$	penalty factor with respect to damping
$\boldsymbol{\omega}$	angular velocity
$\tilde{\boldsymbol{\omega}}_{n-1}$	skew-symmetric matrix of the angular velocity vector
$\tilde{\boldsymbol{\omega}}_{n-1,n}$	skew-symmetric matrix of the relative angular velocity vector
$\dot{\boldsymbol{\omega}}_{n}$	angular acceleration of body n
$\boldsymbol{\Phi}$	Jacobian matrix of constraint equations
$\boldsymbol{\Phi}_{,z}$	Jacobian matrix of the constraint equations with respect to the relative coordinates, z
$\dot{\boldsymbol{\Phi}}$	first derivative of Jacobian matrix of constraint equations
$\ddot{\boldsymbol{\Phi}}$	second derivative of Jacobian matrix of constraint equations
λ	Lagrange multiplier
δ	the operator of variation
δW	virtual work

Abbreviations

ARPACK	Arnoldi Package
CSC	Compressed Sparse Column
CSPARSE	C code Sparse Package
dis	displacement
DOFs	Degrees of freedom
kin	kinetic
nz	number of zero entry
nnz	number of non-zero entry
NCSS	Normet Concrete Spray System
pot	potential
$pris$	prismatic joint
rev	revolute joint
UMFPACK	Unsymmetric MultiFrontal Package

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

652. TALVITIE, JOONAS. Development of measurement systems in scientific research: Case study. 2015. Diss.

653. ZUBEDA, MUSSA. Variational ensemble kalman filtering in hydrology. 2015. Diss.

654. STEPANOV, ALEXANDER. Feasibility of industrial implementation of laser cutting into paper making machines. 2015. Diss.

655. SOKOLOV, MIKHAIL. Thick section laser beam welding of structural steels: methods for improving welding efficiency. 2015. Diss.

656. GORE, OLGA. Impacts of capacity remunerative mechanisms on cross-border trade. 2015. Diss.

657. AURINKO, HANNU. Risk assessment of modern landfill structures in Finland. 2015. Diss.

658. KAIJANEN, LAURA. Capillary electrophoresis: Applicability and method validation for biorefinery analytics. 2015. Diss.

659. KOLHINEN, JOHANNA. Yliopiston yrittäjämäisyyden sosiaalinen rakentuminen. Case: Aalto-yliopisto. 2015. Diss.

660. ANNALA, SALLA. Households’ willingness to engage in demand response in the Finnish retail electricity market: an empirical study. 2015. Diss.

661. RIABCHENKO, EKATERINA. Generative part-based Gabor object detector. 2015. Diss.

662. ALKKIOMÄKI, VILLE. Role of service and data reuse in enterprises. 2015. Diss.

663. VÄNTSI, OLLI. Utilization of recycled mineral wool as filler in wood plastic composites. 2015. Diss.

664. KLEMOLA, KATJA. Tuottavuuden, vaikuttavuuden ja kustannusvaikuttavuuden arviointi alueellisesti integroiduissa sosiaali- ja terveyspalveluissa – palvelujen käyttöön perustuva malli ja esimerkkejä. 2015. Diss.

665. HEROLD, KRISTIINA. Impact of Word-of-Mouth on consumer decision-making: An information processing perspective in the context of a high-involvement service. 2015. Diss.

666. OLABODE, MUYIWA. Weldability of high strength aluminium alloys. 2015. Diss.

667. VANHALA, ERNO. The role of business model in computer game development organizations. 2015. Diss.

668. SALAMPASIS, DIMITRIOS. Trust-embedded open innovation: Towards a human-centric approach in the financial industry. 2015. Diss.

669. DE SMET, DIETER. Innovation ecosystem perspectives on financial services innovation. 2015. Diss.

670. PORRAS, PÄIVI. Utilising student profiles in mathematics course arrangements. 2015. Diss.

671. SALMINEN, JUHO. The role of collective intelligence in crowdsourcing innovations. 2015. Diss.

672. ROSAS, SAILA. Co-operative acquisitions – the contextual factors and challenges for co-operatives when acquiring an investor-owned firm. 2015. Diss.

673. SINKKONEN, TIINA. Item-level life-cycle model for maintenance networks – from cost to additional value. 2015. Diss.

674. TUUNANEN, JUSSI. Modelling of changes in electricity end-use and their impacts on electricity distribution. 2015. Diss.

675. MIELONEN, KATRIINA. The effect of cationic-anionic polyelectrolyte multilayer surface treatment on inkjet ink spreading and print quality. 2015. Diss.

676. OMAJENE, JOSHUA. Underwater remote welding technology for offshore structures. 2015. Diss.

677. NUUTINEN, PASI. Power electronic converters in low-voltage direct current distribution – analysis and implementation. 2015. Diss.

678. RUSATSI, DENIS. Bayesian analysis of SEIR epidemic models. 2015. Diss.

679. STRAND, ELSI. Enhancement of ultrafiltration process by pretreatment in recovery of hemicelluloses from wood extracts. 2016. Diss.

680. TANNINEN, PANU. Press forming of paperboard – advancement of converting tools and process control. 2015. Diss.

681. VALTONEN, PETRI. Distributed energy resources in an electricity retailer's short-term profit optimization. 2015. Diss.

682. FORSSTRÖM-TUOMINEN, HEIDI. Collectiveness within start up-teams – leading the way to initiating and managing collective pursuit of opportunities in organizational contexts. 2015. Diss.

683. MAGUYA, ALMASI. Use of airborne laser scanner data in demanding forest conditions. 2015. Diss.

684. PEIPPO, JUHA. A modified nominal stress method for fatigue assessment of steel plates with thermally cut edges. 2015. Diss.

685. MURASHKO, KIRILL. Thermal modelling of commercial lithium-ion batteries. 2016. Diss.

686. KÄRKKÄINEN, TOMMI. Observations of acoustic emission in power semiconductors. 2016. Diss.

687. KURVINEN, EMIL. Design and simulation of high-speed rotating electrical machinery. 2016. Diss.

688. RANTAMÄKI, JUKKA. Utilization of statistical methods for management in the forest industry. 2016. Diss.

689. PANOVA, YULIA. Public-private partnership investments in dry ports – Russian logistics markets and risks. 2016. Diss.